A Generalized^2 Linear^2 Models módszer implementálása Octave rendszerben

Fülöp András
PTi MSc - Mesterséges Intelligencia szakirány

Konzulens:
Dr. Ispány Márton
Egyetemi Docens

2009. május 8.

Köszönetnyilvánítás

Szeretnék köszönetet mondani debreceni tanáraimnak, akik nagyban segítették munkámat, azáltal, hogy megfelelő tudást adtak át nekem. Külön szeretném kiemelni konzulensemet, Dr. Ispány Mártont, valamint Dr. Varterész Magdát, Jeszenszky Pétert, és Dr. Barczy Mátyást.

Szeretném megköszönni továbbá kecskeméti tanáraimnak is a megfelelő „lökést" ahhoz, hogy sikerült eljutnom eddig a pontig, szeretném kiemelni Dr. Pintér István Tanár Urat, akitől nagyon sokat tanultam az ott eltöltött évek során.

Külön kell szólnom Dr. Török Leventéről, akit személyes mentoromnak tekintek, és köszönöm azt a sok segítséget amit eddig, valamint a diplomamunkámhoz nyújtott.

Szeretném megköszönni szüleimnek, barátnőmnek a támogatást, noszogatást, biztatást ami elengedhetetlen volt ahhoz, hogy ez a dolgozat megszülessen, valamint hogy képesek voltak tolerálni a munka alatt.

Köszönöm barátaimnak és testvéremnek, akik segítettek néha kilépni a munka feszült menetéből, és segítettek kikapcsolódni.

Szimbólumok jegyzéke

2. fejezet.

X	statisztikai változó
Ω	eseménytér
ϑ	paraméter
Θ	paramétertér
F	eloszlásfüggvény
F^*	empirikus eloszlásfüggvény
$\overline{X}$	empirikus átlag
σ^2	elméleti szórásnégyzet
s^2	empirikus szórásnégyzet
T	statisztika
$\hat{\vartheta},\ g(\vartheta)$	ϑ paraméter becslése
L	maximum likelihood becslés

3. fejezet.

X	adatmátrix
Y	főkomponens
$\mathbb{E}$	várható érték
λ	sajátérték
v	sajátvektor
σ	szinguláris érték
$u,\ v$	szinguláris vektor
$\parallel M \parallel_F$	M mátrix Frobenius-normája

4. fejezet.

μ	átlag (Y várható értéke)
η	lineáris prediktor
θ	paraméter
y	megfigyelés
$f(y,\theta)$	y sűrűségfüggvénye θ paraméterrel
$D*$	skálázott eltérésfüggvény
$g(.)$	link függvény
ϕ	diszperziós paraméter
$\theta(\mu)$	kanonikus link
Φ	kumulált eloszlásfüggvény
λ	paraméter

5. fejezet.

$A,\ B,\ U,\ V$	paramétermátrix
$f,\ g,\ h$	link függvények
D_F	Bregman távolság
$F,\ G,\ H$	veszteségvüggvény
α	tanulási együttható

Tartalomjegyzék

1. fejezet

Bevezetés

Az információs társadalom tagjaként annyi információ éri a mindennapok emberét, hogy csak annak töredékét dolgozza fel. Agya megpróbálja szelektálni a releváns információkat, már amennyiben képes erre egy olyan manipulatív média mellett, mint napjainké.

Az információ kinyerése nem csak individuumként fontos feladat, statisztikusként, szociológusként, informatikusként gyakran találkozik akkora adattáblákkal melyek feldolgozásával nem csak az embereknek, de a számítógépeknek is meggyűlik a baja. A felesleges adatok kiszűrésére külön tudományág is épült, az adatbányászat.

Informatikus hallgatóként azért vonzó a feladat, mert gyakran találkozom tanuló algoritmusokkal. Ezek feladattól függően akár óriási állapottereket is átszámolhatnak minden iteráció során. Jelenlegi számítógépem nem csak a http://www.top500.org/ oldalon listázott gépek számítási teljesítményéhez képest gyenge, de egy átlagos mai konfigurációhoz képest is van szégyenkezni valója. Ezért a többet ésszel, mint erővel elv motiválta munkámat.

Dolgozatomban egy olyan eljárást implementáltam, mely segítségével a fent vázolt probléma nagymértékben redukálható, hiszen egy nagy adatmátrix redukált komponensű becslésével a számítási igény csökken. Erre létezik több megoldás is, ám az adatok sokfélesége miatt egy általános megoldás után néztem, így akadtam rá a G^2L^2M módszerre, mely több modellt is magába foglal.

Úgy gondolom, hogy az információs társadalom tagjait az szolgálja legjobban, ha az egyes fejlesztésekhez nem csak egy korlátozott csoport fér hozzá, ezért egy olyan matematikai rendszert választottam, melynek bármely platformra létezik implementációja, azok működése nem tér el egymástól, és végül nem utolsó sorban ingyenesen hozzáférhető. Ezeket a kritériumokat a GNU Octave rendszer mind magába foglalta, és használata is kényelmes. Az implementált algoritmus kompatibilis a MatLab rendszerrel is, így az azzal rendelkezők is felhasználhatják a programomat.

Diplomamunkám második fejezetében a becsléselméleti alapfogalmakat tisztázom, harmadik fejezétben pedig alapvető dimenziócsökkentő eljárásokat írok le. A negyedik fejezet során a GLM módszert írom le, majd az ötödik fejezetben a címben megjelölt módszer leírását közlöm. A hatodik fejezetet az algoritmusnak, illetve a felhasznált programozói felületnek szentelem. Hetedik fejezetben összefoglalom a dolgozatom során elért eredményt, és javaslatot teszek a továbblépésre is.

Minden fejezet végén található egy rövid összegzés a fejezetben olvasottakról, valamint a fejezet megírásához felhasznált irodalmak listája, és az ahhoz fűzött esetleges megjegyzések.

2. fejezet

Becsléselméleti alapfogalmak

Ebben a fejezetben a dolgozat megértéséhez szükséges alapfogalmakat ismertetem, melyeket a későbbiekben fel is használok. Az egyetemi tanulmányaim során az [1] könyvet használtam, így számomra ez a fogalomkör, és jelölésrendszer a megszokott, emiatt ezt a fejezetet is ennek alapján készítettem el. Az első részben néhány ismert statisztikai fogalmat definiálok, majd rátérek a becsléselméleti fogalmak ismertetésére, a becslési módszerek közül a maximum likelihood becslést kiemelve.

2.1. Statisztikai alapfogalmak

A statisztikai alapfogalmak ismertetését azért tartottam fontosnak, mert fontos becsléselméletben felmerülő definíciók támaszkodnak ezekre az ismeretekre. Ebben a részben a minta, mintatér, statisztikai mező, átlag, és szórásnégyzet definícióit írom le, majd magát a statisztikát mint fogalmat is definiálom.

2.1.1. Minta

2.1.1.1. Minta, minta realizáció

Amikor valamilyen jelenséget matematikai nézőpontból figyelünk meg, akkor először megpróbáljuk megszámolni, megmérni. A megmért mennyiség jellemez egy megfigyelt jelenséget. Amennyiben ezt statisztikai szemszögből nézzük, akkor ez egy statisztikai változó, melyet X-szel jelölünk. Ilyen statisztikai változó lehet például egy egyetemista naponta számítógép előtt töltött ideje percekben mérve. Ezt a mérést megismételjük több egymást követő napon is, így megkapjuk X_1-et, X_2-t, egészen X_n-ig. Ezt a megfigyelés sorozatot nevezzük mintának. „Ezt a megfigyelés halmazt nem egy szám n-esnek tekintjük, hanem olyan objektumnak, amely magába sűríti a megfigyelések eredményeként adódó összes lehetséges szám n-est.” [1] Ebből következik, hogy a megfigyelés sorozat elemei is statisztikai változók.

Definíció: Az $X_1, \ldots, X_n$független, azonos eloszlású valószínűségi változókat *minta*nak nevezzük. Rögzített $\omega \in \Omega$ esetén az $x_1 = X_1(\omega), \ldots, x_n = X_n(\omega)$szám n-est *minta realizáció*nak nevezzük. (Itt az Ω a háttérben lévő eseményteret jelöli.)

Megjegyzés:

1. A gyakorlatban mindig minta realizációkat figyelünk meg. Ezek azonban megfigyeléssorozatonként különböznek egymástól. A minta elméleti fogalma az összes realizációt magába foglalja.

2. Ha X egy valószínűségi változó, akkor X-re vett minta alatt az X-szel azonos eloszlású, független $X_1, X_2, \ldots, X_n$ valószínűségi változókat értjük.

3. Ha F egy eloszlásfüggvény, akkor F eloszlásfüggvényű populációból vett minta alatt független, F eloszlásfüggvényű $X_1, X_2, \ldots, X_n$ valószínűségi változókat értünk. (lsd. [1])

2.1.1.2. Statisztikai mező

Amennyiben valószínűségszámításról beszélünk, mindig feltételezünk egy valószínűségi mezőt, mely a valószínűségi változó néhány tulajdonságát írjuk le. Amennyiben X valószínűségi változóról beszélünk, annak ismerjük az eloszlásfüggvényét, F-et, valamint azt, hogy X-et Ω-n értelmezik. A statisztika abban különbözik ettől, hogy itt a feladat pontosan F eloszlásfüggvény (vagy annak ismeretlen paramétereinek) megismerése a megfigyelések segítségével.

Definíció: "Legyen Θ egy nem üres halmaz, és minden $\vartheta \in \Theta$-ra legyen $(\Omega, \mathcal{A}, \mathbb{P}_\vartheta)$ valószínűségi mező. Az $(\Omega, \mathcal{A}, \mathbb{P}_\vartheta)$, $\vartheta \in \Theta$ összességét statisztikai mezőnek nevezzük. Θ-t *paramétertér*nek, elemeit *paraméter*eknek nevezzük.

Az $X_1, X_2, \ldots, X_n$ minta az Ω-n értelmezett a mintaelemek együttes eloszlásfüggvénye pedig $\prod_{i=1}^{n} F_\vartheta(x_i)$, $x_i \in \mathbb{R}$, $i = 1, \ldots, n$. Itt $F_\vartheta(x)$ egyetlen elem eloszlásfüggvénye, a minta együttes eloszlásfüggvénye pedig a függetlenség miatt szorzat alakú. Az F_ϑ eloszlásfüggvény éppen akkor lép fel, amikor a statisztikai mezőn a $\mathbb{P}_\vartheta$ valószínűség aktuális. A gyakorlatban a statisztikai mező a háttérben marad, ténylegesen az F_ϑ eloszlásfüggvénnyel dolgozunk, célunk az ismeretlen ϑ paraméter felderítése." [1]

2.1.1.3. Az empirikus eloszlásfüggvény

Az empirikus eloszlásfüggvényekről akkor beszélünk, amikor olyan feladatot kapunk, mely során ismerjük a mintát, ám annak eloszlását nem. Ilyen esetben az *empirikus eloszlásfüggvény*t keressük. A definíció kimondásához szükségünk van egy másik fogalomra, a rendezett mintára.

Definíció: „Legyen $\omega \in \Omega$ rögzített, jelölje $X_1^*(\omega) \leq X_2^*(\omega) \leq \ldots \leq X_n^*(\omega)$ az $X_1(\omega), X_2(\omega), \ldots, X_n(\omega)$ minta realizáció elemeinek nagyság szerint növekvő permutációját. Az $X_1^* \leq X_2^* \leq \ldots \leq X_n^*$ valószínűségi változókat *rendezett minta*nak nevezzük." [1]

Ezen meghatározást felhasználva, kimondhatjuk az empirikus eloszlásfüggvény definícióját.

Definíció: „Legyen $X_1^*, X_2^*, \ldots, X_n^*$ rendezett minta. Az

$$F_n^*(x) = \begin{cases} 0 & ha\ x \leq X_1^*, \\ \frac{k}{n} & ha\ X_k^* < x \leq X_{k+1}^*,\ k = 1, \ldots, n-1, \\ 1 & ha\ x > X_n^* \end{cases}$$

függvényt *empirikus eloszlásfüggvény*nek nevezzük." [1]

2.1.2. Statisztikák

Egy mintát több módon is jellemezhetünk, azonban a legkézenfekvőbb megoldás, ha ezt a szórás segítségével tesszük meg. Az ilyen jellemzés azonban nagyon megtévesztő lehet, valamint érzékeny a kiugró adatokra is, így egymagában nem elég. Ebben a részben néhány alapvető statisztikát definiálok, melyeket a későbbiekben fel is használok.

2.1.2.1. Empirikus közép

„Legyen $X_1, X_2, \ldots, X_n$ minta X-re. Az

$$\overline{X} = \frac{1}{n}(X_1 + X_2 + \ldots + X_n)$$

valószínűségi változót *empirikus közép*nek (más szóval minta átlagnak) nevezzük.

Tegyük fel, hogy X-nek létezik véges várható értéke: $m = \mathbb{E}X$. Amennyiben m nem ismert, úgy a minta alapján meghatározható $\overline{X}$ várható értéke és szórásnégyzete:

$$\mathbb{E}\overline{X} = \frac{1}{n}\sum_{1}^{n}\mathbb{E}X_i = m,$$

$$\mathbb{D}^2\overline{X} = \frac{1}{n^2}\sum_{1}^{n}\mathbb{D}^2X_i = \frac{\sigma^2}{n},$$

ahol $\sigma^2 = \mathbb{D}^2X$ az elméleti szórásnégyzet.” [1]

2.1.2.2. Empirikus szórásnégyzet

Az

$$s_n^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

mennyiséget *empirikus szórásnégyzet*nek nevezzük. Az empirikus szórásnégyzet alapján következtethetünk X ismeretlen (elméleti) szórásnégyzetére. Ki fog derülni, hogy erre a célra alkalmasabb s_n^2 alábbi módosítását használni. Az

$$s_n^{*2} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

mennyiséget korrigált empirikus szórásnégyzetnek nevezzük.” [1]

2.1.2.3. Statisztika fogalma

„Legyen $X_1, X_2, \ldots, X_n$ minta X-re.

Definíció: Legyen $T : \mathbb{R}^n \to \mathbb{R}^k$ Borel-mérhető[1] függvény. Ekkor a

$$T(X_1, \ldots, X_n)$$

valószínűségi vektorváltozót *statisztikának* nevezzük. A fenti definícióban k rögzített pozitív egész szám, k értéke leggyakrabban 1. Az empirikus közép az egyik legegyszerűbb statisztika, ekkor a T függvény:

$$T(x_1, \ldots, x_n) = \frac{1}{n}(x_1 + \ldots + x_n).$$

Természetesen s_n^2 és s_n^{*2} is statisztikák. F_n^* empirikus eloszlásfüggvény is statisztikának tekinthető...” [1]

[1] Egy $\xi : \mathbb{R} \to \mathbb{R}$ függvényt Borel-mérhetőnek nevezünk, ha $\xi^{-1}(B) \in \mathcal{B}$ minden $B \in \mathcal{B}$ esetén, azaz Borel-halmaz inverz képe Borel-halmaz. [2]

2.2. Becsléselméleti alapfogalmak

Ebben a részben azokat az alapfogalmakat írom le, melyeket a megértés szempontjából fontosnak tartok. A fejezet során a pontbecsléstől indulok ki, majd eljutok a becslések különböző típusaiig.

2.2.1. Pontbecslés

Ahogy fentebb olvasható, a statisztikák feladata egy ismeretlen ϑ paraméter becslése az ismert $X_1, \ldots, X_n$ minta alapján. Feltételezzük, hogy létezik egy ϑ^* valódi érték, amit a $\hat{\vartheta}$-val becslünk, és a minta P_{ϑ^*} eloszlású, de ezt nem ismerjük. A ϑ^* valódi értékét egy olyan függvénnyel próbáljuk megbecsülni, amely az $X_1, \ldots, X_n$ mintáját Θ-ba képzi le. „Így a pontbecslés fogalmához jutunk.

Definíció: Legyen $X_1, \ldots, X_n$ minta az $(\Omega, \mathcal{A}, P_\vartheta)$, $\vartheta \in \Theta$, statisztikai mezőn, $g : \Theta \to \mathbb{R}^k$ egy adott függvény és $T : \mathbb{R}^n \to \mathbb{R}^k$ Borel-mérhető függvény. Ekkor a

$$\hat{g}(\vartheta) = T(X_1, \ldots, X_n)$$

statisztikát a ϑ paraméter g függvénye becslésének nevezzük. Ha speciálisan $\Theta \subset \mathbb{R}^k$ és g az identikus leképzés $\mathbb{R}^k$-n, akkor azt mondjuk, hogy a

$$\hat{\vartheta} = T(X_1, \ldots, X_n)$$

statisztika a ϑ paraméter becslése. Ha a minta elemszámára is utalni akarunk, akkor a T_n, illetve a $\hat{\vartheta}_n$ jelöléssel élünk." [1]

Ezzel a definícióval kapcsolatban több megállapítást is kell tennünk:

1. „Vegyük észre, hogy a definícióban semmit sem követelünk meg g és T kapcsolatáról (csak a dimenzióik egyezését)." [1]

2. „Mind $\hat{\vartheta}$, mind pedig $\hat{g}(\vartheta)$ valószínűségi változó, így értéke realizációnként más és más lehet. Ezzel szemben a ϑ^* valódi paraméterérték konstans, így a $\hat{\vartheta}$ (ill. $\hat{g}(\vartheta)$) becslés egy minta realizáció esetén felvett értéke nagyon messze is eshet a ϑ^* (ill. $g^*(\vartheta)$) valódi paraméterértékétől." [1]

3. „A definícióban az úgynevezett pontbecslés fogalmát vezettük be. Ez azt jelenti, hogy a ϑ paraméter közelítésére a paraméterhalmaz egy, a minta realizációtól függő pontját jelöltük ki. Tekinthetünk ... olyan becsléseket is, amikor egy pont helyett a paraméterhalmaz egy részhalmazát jelöljük ki. Ezeket *halmazbecslés*eknek nevezzük." [1]

Az alábbiakban egy példa segítségével szeretnék bemutatni különböző alakú becsléseket. A paraméter, melyet becsülni szeretnénk, az m ($m = \mathbb{E}X$, és $m \in \mathbb{R}$). Ebben az esetben a becslést

1. lineáris becslésnek nevezzük, ha

$$\hat{m} = \sum_{i=1}^{n} c_i X_i, \qquad c_i \in \mathbb{R},\ i = 1, \ldots, n,$$

alakú;

2. Terjedelemközépnek, ha
$$\hat{m} = (X_1^* + X_n^*)/2,$$
alakú; és

3. Empirikus medián, ha
$$\hat{m} \begin{cases} X_{k+1}^* & ha\ n = 2k+1 \\ (X_k^* + X_{k+1}^*)/2 & ha\ n = 2k \end{cases}$$
alakban adjuk meg. [1]

2.2.2. A becsléselmélet feladata

Egy becslési feladat során, a paraméterekre sok becslés adható. A becslések közül azt kell kiválasztani, amely értéke a legjobban közelít a paraméter valódi értékéhez. „...Egy „jó" T becsléssel szemben megköveteljük az alábbiakat:

1. a T statisztika értékei ϑ (ill. $g(\vartheta)$) valódi értéke körül ingadozzanak,
2. a T statisztika szórása a lehető legkisebb legyen,
3. ha a T statisztika minden minta elemszám esetén értelmezhető, akkor az így kapott T_n, $n = 1, 2, \ldots$ statisztika sorozat konvergáljon a valódi paraméterértékhez." [1]

A szakasz további részében ezen tulajdonságokat definiáljuk.

2.2.2.1. Torzítatlan becslések

Definíció: „A $T : \mathbb{R}^n \to \mathbb{R}^k$ statisztikát a $g(\vartheta)$ *torzítatlan becslés*ének nevezzük, ha minden $\vartheta \in \Theta$ esetén
$$\mathbb{E}_\vartheta T(X_1, \ldots, X_n) = g(\vartheta).$$
Speciálisan, ha $\Theta \subset \mathbb{R}^k$ és g az identikus leképzés, azaz
$$\mathbb{E}_\vartheta T(X_1, \ldots, X_n) = \vartheta$$
minden $\vartheta \in \Theta$ esetén, akkor azt mondjuk, hogy a T statisztika a ϑ paraméter *torzítatlan becslése*. A ϑ paraméter
$$t(\vartheta) = \mathbb{E}_\vartheta T(X_1, \ldots, X_n) - \vartheta$$
függvényét a T *becslés torzításá*nak nevezzük. A T_n becslés sorozatot *aszimptotikusan torzítatlan*nak nevezzük, ha
$$\lim_{n\to\infty} t_n(\vartheta) = 0,$$
ahol t_n a T_n becslés torzítása $n = 1, 2, \ldots$esetén." [1]

Nem minden paraméterre lehet torzítatlan becslést adni, ha mégis, akkor azt a paramétert *becsülhető*nek nevezzük.

2.2.2.2. Becslések hatásossága

Két becslést összehasonlíthatunk az alapján, hogy melyik jobb becslése egy adott paraméternek. A jobb becslést hatásosabb becslésnek nevezzük.

Definíció: „Jelölje $\mathbb{D}^2_\vartheta$ a ϑ paraméter szerinti szórásnégyzetet, azaz $\mathbb{D}^2_\vartheta(X) = \mathbb{E}_\vartheta(X - \mathbb{E}_\vartheta X)^2$. Legyenek T_1 és T_2 a $g(\vartheta)$ véges szórású, torzítatlan becslései[2]. Azt mondjuk, hogy T_1 hatásosabb becslése $g(\vartheta)$-nak T_2-nél, ha minden $\vartheta \in \Theta$ esetén fenáll, hogy

$$\mathbb{D}^2_\vartheta(T_1) \leq \mathbb{D}^2_\vartheta(T_2).$$

Ha $g(\vartheta)$ egy T véges szórású, torzítatlan becslése minden más véges szórású, torzítatlan becslésnél hatátosabb, akkor T-t $g(\vartheta)$ *hatásos* becslésének nevezzük." [1]

2.2.2.3. Konzisztens becslések

Egy adott $X_1, \ldots, X_n$ mintához legyen adott $T_n = T_n(X_1, \ldots, X_n)$, $n = 1, 2, \ldots$ becsléssorozat. [1]

Definíció: „A T_n, $n = 1, 2, \ldots$ becslés sorozatot a ϑ paraméter *(gyengén) konzisztens* becslésének nevezzük, ha $T_n \to \vartheta$ sztochasztikusan, amint $n \to \infty$, azaz ha minden $\varepsilon > 0$ és $\vartheta \in \Theta$ esetén

$$\lim_{n\to\infty} P_\vartheta(|T_n - \vartheta| \geq \varepsilon) = 0$$

teljesül." [1]

[2] „Egy T statisztikát véges szórásúnak fogunk nevezni, ha $\mathbb{D}^2_\vartheta(T)$ véges minden $\vartheta \in \Theta$ esetén." [1]

2.3. A Maximum Likelihood becslés

Ebben a részben egy olyan becslési módszert mutatok be, mely a paramétereket az alapján becsli meg, hogy mikor - mely paraméterérték mellett - lesz a mintánk a legvalószínűbb.

2.3.1. A likelihood függvény

Az általánosított sűrűségfüggvény[3] alternatív elnevezése a likelihood függvény.

A likelihood függvény tárgyalása közben egy másik statisztikai mezőt használok, Ez a minta által indukált statisztikai mező. Formája: $(\mathbb{R}, \mathcal{B}(\mathbb{R}), P_\vartheta^X), \vartheta \in \Theta$. [1]

Definíció: „Legyen az X valószínűségi változó által indukált $(\mathbb{R}, \mathcal{B}(\mathbb{R}), P_\vartheta^X), \vartheta \in \Theta$, statisztikai mező dominált[4] $f(x, \vartheta)$ általánosított sűrűségfüggvényekkel. Tekintsünk egy X-re adott $X_1, \ldots, X_n$ mintát, és jelölje $\mathcal{X} \subset \mathbb{R}^n$ a mintateret. Az

$$L : \mathcal{X} \times \Theta \to \mathbb{R}_+, \quad L(x_1, \ldots, x_n, \vartheta) := \prod_{i=1}^{n} f(x_i, \vartheta)$$

függvényt az $X_1, \ldots, X_n$ mintához tartozó *likelihood függvény*nek nevezzük." [1]

A fenti definíció által leírt függvény a gyakorlatban egy minta együttes eloszlását jelöli diszkrét esetben, sűrűségfüggvényt ϑ paraméterrel abszolút folytonos esetben, annyi különbséggel, hogy ϑ lesz változó, $x_1, \ldots, x_n$ pedig paraméter. [1]

2.3.2. Likelihood becslés

Definíció: „Legyen $L : \mathcal{X} \times \Theta \to \mathbb{R}_+$ az $X_1, \ldots, X_n$ mintához tartozó likelihood függvény. A $\hat{\vartheta} : \mathcal{X} \to \Theta$ statisztikát a ϑ paraméter maximum likelihood becslésének nevezzük, ha $\hat{\vartheta}$ globális maximumhelye a likelihood függvénynek, azaz

$$L(x_1, \ldots, x_n; \hat{\vartheta}(x_1, \ldots, x_n)) \geq L(x_1, \ldots, x_n; \vartheta)$$

minden $\vartheta \in \Theta$ és $(x_1, \ldots, x_n) \in \mathcal{X}$ esetén." [1]

A maximum likelihood becslés nem egyértelmű. Ilyen esetben a maximumhelyek közül választhatunk úgy, hogy $\hat{\vartheta}$ a minta realizációk mérhető függvénye legyen. [1]

2.3.3. Likelihood egyenlet

Ismert, hogy a logaritmus szigoran monoton növekvő függvény, így L maximumhelyének meghatározása helyett dolgozhatunk az alábbi kifejezéssel:

$$\ln L(x_1, \ldots, x_n; \vartheta) = \sum_{i=1}^{n} \ln f(x_i, \vartheta).$$

Ezt *loglikelihood függvény*nek is szokás nevezni. Tegyük fel, hogy $\Theta \subset \mathbb{R}$, ebben az esetben elvégezhető a

$$\sum_{i=1}^{n} \frac{\partial}{\partial \vartheta} \log f(x_i, \vartheta) = 0$$

[3] A Radon-Nikon deriváltakat általánosított sűrűségfüggvénynek nevezzük. Alakja: $f(x, \vartheta) := \frac{\partial P_\vartheta}{\partial \mu}(x)$, általánosított alakja: $\prod_{i=1}^{n} f(x_i, \vartheta)$. [1]

[4] lásd [1] 52. oldal.

likelihood egyenlet[5] megoldása segítségével. Tegyük fel, hogy $\log f(x,\vartheta)$ folytonosan differenciálható Θ-n, ahol a likelihood egyenletnek létezik $\hat{\vartheta} = \hat{\vartheta}(x_1,\ldots,x_n)$ egyértelmű megoldása, ahol az $\log f(x,\vartheta)$ kétszer differenciálható ϑ szerint, és

$$\frac{\partial^2 \log L}{\partial \vartheta^2} \mid_{\vartheta=\hat{\vartheta}} = \sum_{i=1}^{n} \frac{\partial^2}{\partial \vartheta^2} \log f(x_i,\vartheta) < 0$$

minden $(x_1,\ldots,x_n) \in \mathcal{X}$ esetén. Ebben az esetben a lokális maximum egyben globális maximum is.

Több becsülendő paraméter esetén elegendő az alábbi

$$\sum_{i=1}^{n} \frac{\partial}{\partial \vartheta_j} \log f(x,\vartheta_1,\ldots,\vartheta_k) = 0, \qquad j = 1,\ldots,k$$

likelihood egyenletrendszert megoldani. [1]

2.3.4. Likelihood becslés néhány nevezetes esetben

2.3.4.1. Normális eloszlás

Normális eloszlás esetén a becslésünk az átlag és a szórásnégyzet statisztikák lesznek, melyekkel m és σ^2 paramétereket becsültük. A minta $X_1,\ldots,X_n$, eloszlásfüggvényét felírva:

$$f(x,m,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}},$$

majd behelyettesítve a loglikelihood függvénybe:

$$\log L(x_1,\ldots,x_n;m,\sigma^2) = -\frac{n}{2}\log 2\pi - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-m)^2.$$

Az alábbi egyenletrendszert megoldva, az $\hat{m} = \overline{X}$ és $\hat{\sigma^2} = s_n^2$ eredményekhez jutunk.

$$\frac{\partial}{\partial m}\log L(x_1,\ldots,x_n;m,\sigma^2) = \frac{1}{\sigma^2}\sum_{i=1}^{n} x_i - nm = 0,$$

$$\frac{\partial}{\partial \sigma^2}\log L(x_1,\ldots,x_n;m,\sigma^2) = -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{i=1}^{n}(x_i-m)^2 = 0.$$

[1]

[5] a továbbiakban a log függvény alatt is az e alapú logaritmust értem.

2.3.4.2. Poisson eloszlás

Az $X_1, \ldots, X_n$ minta λ paraméterű Poisson eloszlását az alábbi módon becsüljük:

Az eloszlást

$$f(k, \lambda) = \frac{\lambda^k}{k!} e^{-\lambda}, \qquad k = 0, 1, \ldots, \qquad \lambda > 0,$$

behelyettesítjük a loglikelihood függvénybe, így kapjuk a

$$\log L(x_1, \ldots, x_n; \lambda) = \log \lambda \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \log x_i! - n\lambda$$

egyenletet, ebből kapjuk a

$$\frac{1}{\lambda} \sum_{i=1}^{n} x_i - n = 0$$

likelihood egyenletet, melyből

$$\frac{\partial^2}{\partial \lambda^2} \log L(x_1, \ldots, x_n; \lambda) = -\frac{1}{\lambda^2} \sum_{i=1}^{n} x_i < 0,$$

amiből a $\hat{\lambda} = \overline{X}$ becslést kapjuk. [1]

2.3.4.3. Exponenciális eloszlás

Az $X_1, \ldots, X_n$ minta λ paraméterű exponenciális eloszlását az alábbi módon becsüljük:

Az eloszlásfüggvényt

$$f(x, \lambda) = \lambda e^{-\lambda x}, \qquad x > 0, \qquad \lambda > 0,$$

behelyettesítjük a loglikelihood függvénybe, így kapjuk a

$$\log L(x_1, \ldots, x_n; \lambda) = n \log \lambda - \lambda \sum_{i=1}^{n} x_i$$

egyenletet, ebből kapjuk a

$$\frac{\partial^2}{\partial \lambda^2} \log L(x_1, \ldots, x_n; \lambda) = -\frac{n}{\lambda^2} < 0,$$

egyenletet, mely a maximumhely ellenőrzéséhez szolgál, melyből

$$\frac{n}{\lambda} - \sum_{i=1}^{n} x_i = 0,$$

amiből $\hat{\lambda_n} = \frac{1}{\overline{X}}$ becslést kapjuk. [1]

2.4. Összegzés

A fejezet során bemutattam, definiáltam azokat az alpvető statisztikai fogalmakat, melyeket fontosnak tartottam ahhoz, hogy a dolgozatom biztos alapokkal rendelkezzen, amire az elkövetkező fejezetek építkezhetnek. A következő fejezet a főkomponens-analízist tárgyalja, mely már épít erre a fejezetre.

2.5. Bibliográfiai megjegyzések a fejezethez

[1] István Fazekas (szerk.). *Bevezetés a matematikai statisztikába.* II-III. fejezet. Kossuth Egyetemi Kiadó, 1997

[2] István Fazekas. *Valószínűségszámítás (Számítógépes segédlet).* IV. fejezet, 1. Debreceni Egyetem, 2000.

[3] MTA MMSz. *A maximum likelihood becslésről.* http://www.muszeroldal.hu/assistance/ml.pdf, 2009.04.19.

[4] Sándor Kabos. *Statisztika II. Becslés.* http://kabos.web.elte.hu/matstat/stat5e.pdf. 2009.04.19.

3. fejezet

Dimenziócsökkentési eljárások

Diplomamunkám címe is mutatja, hogy egy dimenzócsökkentő eljárásról írom dolgozatomat, és úgy gondoltam, hogy nem kaphat teljes képet egy témáról senki, ha nem vizsgál meg, és nem hasonlít össze több különböző eljárást.

Ebben a fejezetben két dimenziócsökkentő eljárást mutatok be, a főkomponens-analízist, valamint a szinguláris felbontást.

3.1. Főkomponens-analízis

Ebben a részben a főkomponens analízist (Principal Component Analysis, PCA) írom le, mivel a későbbiekben szükség lesz ennek a módszernek az ismeretére.

A főkomponens analízis egy olyan többváltozós statisztikai módszer, melynek lényege, hogy egy olyan olyan kevesebb változóból álló változóhalmazt keres úgy, hogy a tárolt információtartalom a lehető legkisebb mértékben csökkenjen.

A fejezet első részében definiálom a főkomponens-analízist, majd bemutatom, hogyan lehet a főkomponens becslésére használni a módszert.

3.1.1. Meghatározás

Legyen X p-dimenziós véletlen vektor, D pozitív definit szórásmátrixszal, ahol D spektrálfelbontása $D = V\Lambda V^T$, ahol Λ olyan diagonális mátrix, melynek főátlójában nagyság szerint rendezett $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p > 0$ sajátértékek állnak, V ortonormált, és oszlopvektorai, $v_1, \ldots, v_p$ a sajátértékeknek[1] megfelelően rendezett sajátvektorok. [5]

Definíció: „Az $Y = V^T X$ véletlen vektort az X főkomponens vektorának nevezzük, az Y vektor i-edik komponense pedig az X i-edik főkomponense.” [5]

Az alábbi állításokat tehetjük[2] a definícióhoz:

- A főkomponensek korrelálatlanok, mert $\mathbb{E}Y = 0$, és $\mathbb{E}YY^T = \Lambda$.
- Egy $p \times p$ méretű, ortonormált W mátrixhoz tartozó X és WX főkomponens vektora megegyezik, vagyis ez a vektor az elforgatással nem változik[3].

[1] A sajátértékek, és sajátvektorok definícióját a 3.2.1.1 részben írom le.
[2] Az állítások a [5] forrásból származnak, bizonyításuktól eltekintek.
[3] A bizonyítás megtalálható [5] 355. oldalán.

- A főkomponens analízisre nem igaz, az az állítás, hogy skálainvariáns, azaz ha a mértékegységet megváltoztatjuk, ami egy $X \mapsto \Delta X$[4] transzformációt jelent.
- „Tétel[5]: Az r-edik főkomponens, Y_r, olyan valószínűségi változó, amelynek szórása maximális az összes olyan valószínűségi változó között, amely
  1. $b^T X$ alakúak, ahol $b \in \mathbb{R}^p$ és $\| b \| = 1$;
  2. korrelálatlan az $Y_1, \ldots, Y_{r-1}$ főkomponensekkel." [5]
- „Tétel[6]: Az X valószínűségi vektorváltozó négyzetes középben vett legjobb k-dimenziós becslése az a vektor, amelynek az első k koordinátája az első k főkomponens, a többi pedig 0." [5]

3.1.1.1. Főkomponens-analízis geometriai értelemben

A főkomponens-analízis főkomponenseit egy olyan tengelynek lehet elképzelni, mely úgy fordul, hogy az elhagyni kívánt változó (amely itt koordináta-tengelyt jelent) szórása a lehető legnagyobb legyen.

Definíció: „Egy véges pontrendszer i-edik főtengelye az az Y_i egyenes, amely merőleges az $Y_1, \ldots, Y_{i-1}$ főtengelyekre, és az ilyen egyenesek közül az Y_i mentén a legnagyobb a pontok szórása, azaz ha a pontrendszer pontjait levetítjük az Y_i egyenesre, akkor a képek origótól vett távolságainak tapasztalati szórásnégyzete a maximális." [5]

Tehát a módszer nem más, mint egy bázistranszformáció, melynek új báziselemei az $\{Y_1, Y_2, \ldots, Y_p\}$ főtengelyek. A minta i.-ik főkomponens vektora a mintaelemek, új bázis szerinti, i-edik koordinátái. [5]

3.1.2. Főkomponensek becslése normális eloszlású mintából

Legyen adott egy $\mathcal{N}_p(0, D)$ paraméterű normális eloszlásból vett minta, $X_1, X_2, \ldots, X_n$, ahol $n > p$ teljesül. A becsléshez D kovarianciamátrix sajátértékeinek, és sajátvektorainak becslésére van szükség. Tegyük fel, hogy D teljes rangú, így maximum likelihood becslése:

$$\hat{D} = S_n = \frac{1}{n} \sum_{i=1}^{n} (X_i - \overline{X})(X_i - \overline{X})^T$$

lesz. [5]

A következő két kijelentést írhatjuk le ezzel kapcsolatban:

- „Ha D sajátértékei különbözőek, akkor D sajátértékeinek és sajátvektorainak maximum likelihood becslései éppen a $\hat{D}$ empirikus kovariancia mátrix megfelelő sajátértékei, és sajátvektorai." [5]
- „Tétel: Tegyük fel, hogy a D mátrix különböző sajátértékei $\lambda_1 > \cdots > \lambda_n > 0$, multiplicitásaik rendre $q_1, \ldots, q_r$ ($\sum_{i=1}^{r} q_i = p$). Ekkor
  1. A λ_i sajátérték maximum likelihood becslése:

$$\frac{1}{q_i} \sum_{j=q_1+\cdots+q_{i-1}+1}^{q_1+\cdots+q_i} \hat{\lambda_j}, \qquad i = 1, 2, \ldots, r,$$

[4] Δ diagonális mátrix
[5] A bizonyítás megtalálható [5] 355. oldalán.
[6] A bizonyítás megtalálható [5] 357. oldalán.

ahol $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_p > 0$ a $\hat{D}$ sajátértékei.

2. A D sajátvektorainak a maximum likelihood becslései a $\hat{D}$ empirikus kovarianciamátrix megfelelő sajátvektorai.„[5]

3.1.3. Példa főkomponens-analízisre

Egy kis előretekintésként, az octave segítségével bemutatok egy főkomponens analízist, lépésről-lépésre. Az octave beépített függvényei segítségével ez a feladat nagyon egyszerű, hiszen minden lépésre létezik beépített függvény:

- létrehozunk egy normális eloszlású, 5000 elemű mintát tartalmazó adatmátrixot:

```
X = randn(5000,2);
```

- egy adatmátrix kovarianciamátrixának számítása:

```
cov_X = cov(X);
```

- Ezek után a sajátértékek, sajátvektorok kiszámítása, sorbarendezése egy függvénnyel egyszerűen megoldható:
  - a függvény definiálása: a visszatérési érték(ek) /itt mátrix lesz/ = függvénynév (az átadott paraméterrel /jelen esetben kovarianciamátrixot vár/):

    ```
    function [V,LAMBDA] = pc_compute (X)
    ```

  - V-be a sajátvektorok, LAMBDA-ba a sajátértékek beírása, eig() függvény segítségével :

    ```
    [V,LAMBDA] = eig (X);
    ```

  - sajátértékek sorrendbe rendezése:

    ```
    [LAMBDA,S] = sort(diag(LAMBDA));
    ```

  - majd a sajátvektorokat rendezése sajátértékek alapján:

    ```
    V = V(:,S);
    ```

  - Függvény definíciójának vége:

    ```
    endfunction
    ```

- Az utolsó lépésként már csak meg kell el kell döntenünk, hogy mely vektort akarjuk elhagyni. Jelen esetben a második oszlop tartalmazza a kevésbé fontos adatokat, ezeket elhagyhatjuk.
  - Így először meghívjuk a függvényt,

    ```
    [V,LAMBDA] = pc_compute(cov_X);
    ```

  - Majd csak az első oszlopot hagyjuk meg:

    ```
    uj_X = V(:,1);
    ```

Természetesen ezt két dimenzió esetén butaság volna megtenni, de el lehet képzelni olyan magas dimenziószámmal is az adott feladatot, hogy értelmes legyen a feladat. Lefuttatva X = (5000, 100) paraméterrel, az utolsó sajátérték körülbelül 0,75 értékű, a legelsőé pedig körülbelül 1,28. A felhasználótól függ, hogy mennyire fontos számára a dimenziócsökkentés, és hány komponenst hagy el.

3.1.4. Összegzés

Ebben a részben a főkomponens-analízist mutattam be, amit a következő fejezetben bemutatott módszerrel hatékonyabban is végre lehet hajtani, ki lehet bővíteni, viszont fontosnak tartottam, hogy ismertessem az eredeti formáját is. A következőkben egy újabb módszert ismertetek, amit összevetek a fent leírt módszerrel is.

3.2. A szinguláris felbontás

A szinguláris felbontás egy olyan dimenziócsökkentési eljárás, mely klasszikus lineáris algebrai módszereket használ. Az eljárás során egy M mátrixból állítunk elő egy $\hat{M}$ mátrixot, melynek értékei az M mátrix értékeihez közeliek.

A szinguláris felbontáshoz szükséges néhány alapfogalmat írok le, majd azután térek az eljárás leírására, és a dimenziócsökkentési módszer taglalására. A fejezet végén egy rövid példa segítségével bemutatom az eljárást.

3.2.1. Kapcsolódó alapfogalmak

3.2.1.1. Sajátérték, sajátvektor

A módszer ismeretéhez szükséges tudni, hogy mit nevezünk egy mátrix sajátértékének, valamint sajátvektorának. Röviden:

- M mátrixhoz tartozó λ számot, melyre

$$|M - \lambda I| = 0$$

  teljesül, ahol I egységmátrix, az M mátrix sajátértékének;

- M mátrixhoz tartozó v vektort, melyre

$$Mv = \lambda v$$

  teljesül, M mátrix sajátvektorának nevezzük. [10]

3.2.1.2. Szinguláris érték, szinguláris vektor

Egy $n \times m$ dimenziójú, M mátrix szinguláris értéke az a nemnegatív σ szám, szinguláris vektorai pedig azok a nemnulla u, v vektorok, melyekre

$$Mv = \sigma u,$$

$$M^T u = \sigma v$$

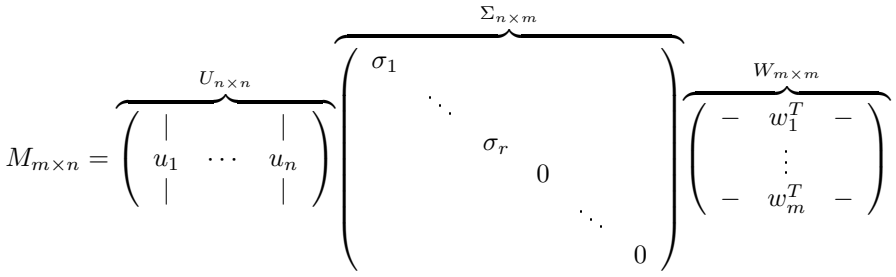
teljesül. [10]

3.2.1.3. Ortonormáltság

Norma. Egy $v \in \mathbb{R}$ vektor 2-normája nem más, mint a hossza, vagyis $\| v \|_2 = \sqrt{\sum_i v_i^2}$, ezt kibővítve, $M \in \mathbb{R}^{n \times m}$ mátrix 2-normája, a Frobenius-norma, számítási módja:

$$\| M \|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} M_{i,j}^2}.$$

[9]



3.1. ábra. A szinguláris felbontás sematikus vázlata. [9, 10]

Ortogonalitás. Ortogonális nevezünk egy $U \in \mathbb{R}^{n\times n}$ mátrixot, amennyiben oszlopai ortogonális rendszert alkotnak, tehát $U^T U = I_n$, ahol I_n az $n \times n$ méretű egységmátrix.
Ha egy ortogonális mátrixszal megszorzunk egy vektort, akkor az nem más, mint egy lineáris transzformáció, egy elforgatás, amely nem változtatja a vektor hosszát, azaz

$$\| Ux \|_2 = \| x \|_2,$$

ahol $U \in \mathbb{R}^{n\times n}$mátrix, és $x \in \mathbb{R}^n$ adott.
$M \in \mathbb{R}^{n\times m}$ mátrixra kiterjesztve, $U \in \mathbb{R}^{n\times n}$ és $VW \in \mathbb{R}^{m\times m}$ ortogonális mátrixokra teljesül, hogy

$$\| UMW^T \|_F = \| M \|_F .$$

[9]

3.2.2. Meghatározás

Definíció: „Ha M valós elemű, n-edrendű négyzetes mátrix, akkor létezik olyan valós ortogonális U és W mátrix, hogy

$$M = U\Sigma W^T$$

ahol Σ diagonálmátrix elemei M szinguláris értékei, az U és W ortogonális mátrixok pedig az MM^T illetve az $M^T M$ valós szimmetrikus pozitív szemidefinit mátrixok modálmátrixai[7].” [8]

Az M mátrix nem csak négyzetes lehet, tekinthetünk egy $n \times m$ méretű mátrixot is, a definíció igaz marad, a felbontás így 3.1. ábrán látható alakban jön létre, ez esetben Σ M-mel megegyező méretű, és értékei M szinguláris értékei, növekvő sorrendben rendezve, azaz $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, ahol $\sigma_i = \sqrt{\lambda_i}$, minden más helyen, ahol $i > r$, $\sigma_i = 0$. Ués W^T ortogonális mátrixok, oszlopaik (u_i, és w_i^T) az M mátrix bal-, ill. jobboldali sajátvektorai[8]. [9, 10]

3.2.3. Dimenziócsökkentés szinguláris felbontás segítségével

A dimenziók relevanciáját, fontosságát a szinguláris felbontás esetén a szinguláris értékekkel érzékeltetik, tehát minél kisebb az adott érték, annál jelentéktelenebb. A kisebb szinguláris értékkel rendelkező részek gyakran a zajból származnak, emiatt elhagyhatóak. Azaz, ha n attribútomot $k < n$ dimenzióval szeretnénk leírni, akkor az első k darab elemet hagyjuk meg a felbontásból. A mátrix-

[7] A modálmátrix olyan mátrix, melynek oszlopai sajátvektorok
[8] r a mátrix rangja.

szorzás tulajdonsága miatt a szinguláris felbontás

$$M = U\Sigma W^T = \sum_{i=1}^{n} \sigma_i u_i w_i^T$$

alakban is felírható, ahol az $u_i w_i^T$ szorzat eredménye egy 1 rangú $n \times m$ méretű mátrix, amelyek egyre csökkenő, $\sigma_i$súllyal szerepelnek a végeredményben. Ekkor csak egy $k < r$ első legkisebb súlyú elemmel közelítsük M mátrix elemeit, azaz

$$\hat{M} = \sum_{i=1}^{r} \sigma_i u_i w_i^T = U_k \Sigma_k W_k^T,$$

ahol $\text{rang}(\hat{M}) = \text{rang}(\Sigma_k) = k$. Így csökkentettük a dimenziókat az információ jelentős torzulása nélkül. [9, 10]

3.2.4. Példa Octave-ban

Az alábbiakban a szinguláris felbontás segítségével elvégzett dimenziócsökkentés lépéseit írom le, Octave nyelven.

- Először létrehozok egy $n \times m$ méretű mátrixot exponenciális eloszlású elemekkel, majd néhány oszlop értékét megnövelem normális eloszlásból származó zajjal.
  - Exponenciális eloszlású adatokból 500 dimenziós, 700 megfigyelést tartalmazó mátrix létrehozása:

    ```
    X = rande (700,500);
    ```

  - Zajmátrix létrehozása, az utolsó 200 dimenziót a hibával, a maradékot 0-val feltöltve:

    ```
    H=[zeros(700,300),randn(700,200)];
    ```

  - A mátrix létrehozása:

    ```
    M = X+H;
    ```

- Majd végrehajtjuk rajta az SVD eljárást:

  ```
  [U,S,W] = svd(M,1);
  ```

Amennyiben adott az svd eljárásnak egy második argumentum, akkor megpróbálja elhagyni az elhagyható elemeket. Az eljárás után sikerült elhagyni a 200 elemet, mely normális eloszlással volt terhelve.

3.2.5. Összegzés

A fejezet során bemutattam a szinguláris felbontás menetét, azt a módot, ahogy ez az eljárás a dimenzió csökkentésére használható, majd egy rövid példát hoztam arra, hogy hogyan lehet implementálni az eljárást Octave rendszerben.

Algorithm 3.1 Az ismertetett eljárás eredménye

```
octave:1> X = rande (700,500);
octave:2> H=[zeros(700,300),randn(700,200)];
octave:3> M = X+H;
octave:4> [U,S,W] = svd(M);
octave:5> size(S)
ans=
700 500
octave:6> [U,S,W] = svd(M,1);
octave:7> size(S)
ans =
500 500
```

3.3. A két módszer összekapcsolása

Ebben a részben bemutatom, hogy milyen összefüggést lehet felírni a két módszer között.

A főkomponens analízis definíció szerinti $Y = V^T X$ alakja felírható $X = YV$ alakban is. Ezt összevetve a szinguláris felbontás definíció szerinti, $X = U\Sigma W^T$ alakjával, látható, hogymindkét esetben az X mátrixot alakítottuk át, tehát felírhatjuk a

$$X = U\Sigma W^T = YV$$

egyenlőséget. Így definiált a két módszer közötti kapcsolat. [10]

3.4. Összegzés

A fejezetben két ismert dimenziócsökkentő eljárást mutattam be, és leírtam egy-egy octave nyelven írt példát, majd a két módszer közötti összefüggést írtam le. A továbbiakban egy újabb módszer következik, mely jóval általánosabb, és a fejezetben említett két módszer egy-egy speciális esetet képez benne.

3.5. Bibliográfiai megjegyzések a fejezethez

[5] Fazekas István (szerk.). *Bevezetés a matematikai statisztikába.* X.2. fejezet. Kossuth Egyetemi Kiadó, 1997.

[6] Jonathon Shlens. *A Tutorial on Principal Component Analysis.* http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf. 2009.04.19.

[7] M. Collins, S. Dasgupta, és R. E. Schapire. *A generalization of principal components analysis to the exponential family.* T. G. Dietterich, S. Becker, and Z. Ghahramani (szerk.). Advances in Neural Information Processing Systems, volume 14, Cambridge, MA, 2002. MIT Press.

[8] Rózsa Pál. *Lineáris Algebra és alkalmazásai.* Tankönyvkiadó, 3. kiadás, 1991.

[9] Bodon Ferenc. *Adatbányászati algoritmusok.* http://people.inf.elte.hu/ktk/Infkez1/adatbanyaszat.pdf. 2009.04.10.

[10] Hren Brunó. *Méréstechnikai és kemometriai módszerek fejlesztése a Fourier-transzformációs Infravörös és Raman-spektroszkópiában.* Doktori (PhD) értekezés, Veszprém 2008. http://twilight.vein.hu/phd_dolgozatok/hrenbruno/Hren_Bruno_Ertekezes.pdf. 2009.04.11.

4. fejezet

GLM

Ebben a fejezetben a Generalized Linear Model-leket (Általánosított Lineáris Modelleket) tekintem át, mélységeibe nem hatolva, a model alapjait ismertetve, rálátást adva a módszerre. A témában sok cikk készült, ennek nagy része a [11] könyvre hivatkozik, emiatt úgy gondoltam ennek a fejezetnek ez a könyv a legmegfelelőbb forrás. A fejezet során az általános elképzelés bemutatása után a modell részeit ismertetem.

4.1. Áttekintés

Az általánosított lineáris modell (Generalized Linear Model - GLM) egy olyan statisztikai módszer, amely a likelihood módszeren alapul. A GLM speciális esetei között megtalálhatjuk a lineáris és loglineáris regressziót, logit és probit modelleket. Ezen eseteken túl, még számos más módszert is tartalmaz. Mindezek különböző eloszlású adatokra alkalmazhatóak, ám van egy közös tulajdonságuk: a linearitás.

A módszer ezt a tulajdonságot használja ki úgy, hogy egy általános algoritmust használ minden esetre ahelyett, hogy külön-külön kezelné azokat. Ez az algoritmus a paraméterek értékét becsli meg, és minden esetre azonos az eljárás, csak bizonyos részek változnak esetenként.

Ez a becslés több iterációs lépésen keresztül éri el az ideális állapotot, ehhez segítségül hív különböző összekötő függvényeket (link functions), valamint veszteségfüggvényeket (loss function). Az összetartozó link-, és veszteségfüggvények a különböző eloszlásokhoz tartoznak, így minden eloszlásra ugyanazt az eljárást kell végrehajtani, azzal a különbséggel, hogy a link, ill. a loss function-t az adott feladat szerint kell megválasztani. [11]

4.2. A modellillesztés lépései

A modellillesztés az adatok értelmezését segíti, ennek három elkülönölő szakaszát különböztetjük meg (Box és Jenkins - 1976 alapján, idősorra vonatkoztatták):

1. Modell kiválasztás
2. Paraméterbecslés
3. Predikció

Az alábbiakban e három szakaszt mutatom be, [11] alapján.

4.2.1. Modell kiválasztás

A modell kiválasztás során választunk modellt az adatainkhoz. Ez azért fontos, mert az adatokhoz próbáljuk illeszteni a modellt, ezért az adatok tulajdonságai határozzák meg, mely modell a legmegfelelőbb.

Fontos tudnunk az adatokról, hogy a Generalized Linear Model-lek megkövetelik, hogy azok független megfigyelésekből származzanak. Általánosabban, a megfigyelések függetlenek lehetnek ismert nagyságú blokkokban. Ezen tulajdonság kizárja az idősorok és terek autokorrelációjából származó adatokat. A függetlenségről tett feltevés a lineáris modellekre és a klasszikus regresszióanalízisre is jellemző, és azt módosítás nélkül helyezték az általnosított lineáris módszerek szélesebb osztályába.

A másik feltevés a hiba alakjára vonatkozik, mely szerint a modellben csak egy hibatag létezik. Ez a megszorítás kizárja azon modelleket, melyek több hibataggal renelkeznek. A legkézenfekvőbb példa, mely ennek a kritériumnak „esik áldozatául", a split-plot design, amely két hibataggal, a teljes-ploton belüli eltéréssel, és a plotok-közötti eltéréssel dolgozik.

A fent említett két megkötés a gyakorlatban korántsem „köti meg annyira az alkalmazó kezét", mint ahogy az első ránézésre tűnik. Példának okáért egy autoregresszív modell könnyedén illeszkedhet olyan programot használva, amely kifejezetten hagyományos lineáris modellekre terveztek.

A skála megválasztása az analízishez a modell kiválasztás egy fontos aspektusa. A gyakori választás az Y (az eredeti skála) és a log Y között történik. Arra a kérdésre, hogy „Milyen a jó skála?" az a válasz adható, hogy az attól függ, hogy mire akarjuk használni a skálát. Erre utal a következő idézet is:

> „Azt a kijelentést, mely szerint néha paradoxonnak tekintik, hogy a válasz nem csak a megfigyelésekben, hanem a kérdésben is rejlik, közhelynek kellene tekinteni" [12]

A klasszikus lineáris regresszióanalízisben a jó skála kombinálja az eltérés állandóságát, a hiba várható normalitását, és a rendszeres hatások additivitását. Jelenlegi tudásunk alapján nem feltételezhetjük, hogy létezik ilyen skála. Például diszkrét adatok esetén, ahol a hibát Poisson eloszlással közelítünk, a szisztematikus hatások pedig multiplikatívak. Ebben az esetben $Y^{\frac{1}{2}}$ közelíti az eltérést, $Y^{\frac{2}{3}}$ jobb közelítést ad a normalitásra, szimmetriára, log Y pedig kezeli az additív hatásokat. Nyílvánvaló, hogy nem létezik olyan skála, amely minden téren jól teljesítene.

A GLM bevezetésével a skálázási problémák nagymértékben csökkennek. A variancia normalitása, és állandósága nem követelmény, habár megköveteljük a variancia és az átlag kapcsolatának ismeretét. Habár az additív hatások még mindig fontos elemei a GLM-eknek, de specifikálhatja hogy megállja a helyét transzformált skálaként szükség esetén. A GLMekben az additivitást a remélt válaszok

tulajdonságaiból származtatjuk. Az additivitás az adatokra vonatkoztatva nem más, mint durva közelítés.

A fennmaradó probléma a modellválasztást illetően az magyarázó-változók kiválasztása. A legegyszerűbb mód, amennyiben adott $x_1 \dots x_2$, keressük annak egy részhalmazát, melyet felhasználva a legjobb illeszkedést kapjuk. Ebben a lépésben fontos az ésszerű részhalmaz kiválasztás, hiszen bár 1-1 változó bevonása a részhalmazba jobb illeszkedéssel kecsegtet, a komplexitást nagymértékben növelheti.

4.2.2. Paraméterbecslés (Estimation)

A modell kiválasztása után a paraméterek becslését, és a becslés pontosságát kell meghatároznunk. A GLM esetén ez a folyamat a jóság meghatározásával történik, mely során megnézzük, hogy a modell által generált értékek és az adathalmaz között mekkora különbség van. A paraméterbecslések azok az értékek, melyek minimalizálják az illeszkedés jóságát (általában jóságfüggvényét). A becslések értékeit a paraméterek likelihood és log-likelihood maximalizálása adja. Amennyiben $f(y;\theta)$ a sűrűségfüggvénye, vagy az y megfigyelés valószínűségi eloszlása a θ paraméter által adott, akkor a log-likelihood, az átlag függvényként, $\mu = E(Y)$,

$$l(\mu; y) = \log f(y; \theta).$$

Az $y_1 \dots y_n$ független megfigyelésen alapuló log likelihood nem más, mint az önálló megfigyelések hozzájárulásának összege, így

$$l(\mu; y) = \sum_i \log f_i(y_i; \theta_i),$$

ahol $\mu = (\mu_1, \dots, \mu_n)$. Megfigyelhető, hogy az $f(y;\theta)$ sűrűségfüggvény az y függvénye fix θ esetén, míg a log likelihood a θ függvénye a megfigyelt y értékekhez. Ezen tényből fakad az argumentumok fordított sorrendje.

Vannak előnyei, ha az $l(\mu; y)$ log likelihood helyett az illeszkedés jóságát vizsgáljuk az alábbi lineáris függvénnyel:

$$D^*(y; \mu) = 2l(y; y) - 2l(\mu; y),$$

melyet skálázott eltérésnek nevezünk (scaled deviance). Exponenciális modellek esetén megfigyelhető, hogy az $l(y; y)$ likelihood függvény maximuma egy meghatározott esetben következik be; amikor az illesztett értékek megegyeznek a megfigyelt adattal. Mivel $l(y; y)$ nem függ a paraméterektől, $l(\mu; y)$ maximalizálása megegyezik $D^*(y; \mu)$ minimalizálásával a μ-t figyelembe véve, alávetve a modell által felvetett korlátozásoknak.

A normális-eloszlást feltételező lineáris regresszió modell ismert σ^2 varianciával, egyszeri megfigyelésre

$$f(y; \mu) = \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right),$$

így a log likelihood

$$l(\mu; y) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(y-\mu)^2}{2\sigma^2}.$$

Az y-t μ-re állítva megkaphatjuk az elérhető maximális log likelihood értéket, a

$$l(y; y) = -\frac{1}{2}\log(2\pi\sigma^2),$$

így a skálázott eltérés függvény

$$D^*(y;\mu) = 2\left\{l(y;y) - l(\mu;y)\right\} = \frac{(y-\mu)^2}{\sigma^2}.$$

4.2.3. Jóslás (Prediction)

A predikció azokra a „mi lesz akkor, ha" („what-if") kérdésekre keresi a választ, amelyeket a statisztikai elemzés után tesznek fel. Általánosabban, a predikció azokkal az álltásokkal foglalkozik, melyek azon valószínűségi értékekre vonatkoznak, amelyek a nem megfigyelt eseményeket jellemezik (ezen eseményeknek nem feltétlenül a jövőben kell bekövetkezniük). Például egy országos szintű felmérés után, mely a szívvel kapcsolatos megbetegedésekre irányult, a tipikus „mi lesz akkor, ha" kérdés a „Mi lesz a jósolt megbetegedési arány egy adott városban, melynek az életkori megoszlása megegyezik az országos eloszlással?". Ez a típusú kérdés a standardizálásra jó példa.

Egy másik példa, ha olyan esetet vizsgálunk, mely során egy kvantális dózis válasz vizsgálatot végzünk (orvosi vizsgálat, mely arra irányul, hogy a vizsgált tesztalanyok mekkora hányada reagált mekkora dózisra). Illesztünk egy modellt arra, hogy a dózissal együtt hogyan változik a reagáló tesztalanyok hányada. Az illesztett modell alapján azt mondjuk (jósoljuk), hogy az az adag, melyre a tesztalanyok 50%-a reagált az LD50. Ez megadja a választ arra a kérdésre, hogy „Mi lenne a jósolt adag, ha a reakció 50%-os volna?". A kalibráció szó gyakran elhangzik ilyen esetekben, hogy megkülönböztesse az inverz predikciós problémát, amelyben a válasz adott, és az x-ek várható értékéről teszünk megállapítást, az általánosabb típustól, mely során a szerepek fordítottak.

Hogy használható legyen, a jövendőlt értékeket a precizitással együtt kell felmutatni. Ezt a precizitást, pontosságot, abból a feltevésből számolják, hogy az a felállás, melyből az eredeti megfigyelések származnak változatlan, és az analízis is helyes volt.

4.3. A GLM részei

Az általánosított lineáris modellek a klasszikus lineáris modellek kiterjesztései, így ebből indulunk ki a felépítés során. Legyen y egy n elemű vektor, amely a μ átlagú, független komponensű Y véletlen változót realizálja. A modell szisztematikus részét (ahogy ezt a 4.2.1 részben látható) úgy kapjuk meg, hogy a μ vektort kis $\beta_1 \dots \beta_p$ taggal bővítjük ki. A hagyományos lineáris rendszerekben ez a specifikáció

$$\mu = \sum_{1}^{p} x_j \beta_j \tag{4.1}$$

alakban jelenik meg, ahol a β-k paraméterek, amelyek értékét becsléssel határozhatjuk meg. Ha a megfigyeléseknek i indexet adunk, a modell szisztematikus része

$$E(Y_i) = \mu_i = \sum_{1}^{p} x_{ij} \beta_j; \qquad i = 1, \dots, n \tag{4.2}$$

képlettel adható meg, ahol x_{ij} a j-edik változó értéke az i-edik megfigyelés esetén. Mátrixalakban (ahol μ $n \times 1$, X $n \times p$, és β $p \times 1$ alakú)

$$\mu = X\beta \tag{4.3}$$

formában is írhatjuk, ahol az X a modell mátrixa, a β a paramétervektor. Ezzel a modell szisztematikus részét definiáltuk.

A modell véletlen részéhez feltételezzük a hibák függetlenségét, és állandó varianciáját. Ezeket a tulajdonságokat ellenőrizni kell, amennyiben lehetséges, magán az adathalmazon. Hasonlóan, a szisztematikus rész is feltételezi, hogy a változókról tudjuk, hogyan befolyásolják az átlagot, és ezt a hatást hiba nélkül tudjuk mérni; így ez is ellenőrzésre szorul.

Egy további megszorítás a modellen, ha feltesszük, hogy a hiba állandó σ^2 szórásnégyzetű normális eloszlást követ.

Összefoglalva az eddigieket, az Y alkotói független, normális elszlású, állandó σ^2 varianciájú változók, és

$$E(Y) = \mu = X\beta. \tag{4.4}$$

4.3.1. Az általánosítás

Hogy egyszerűsítsük az átállást az általánosított lineáris modellekre, átrendezzük a (4.4)-es egyenletet, hogy ehhez a három pontból álló megállapításhoz jussunk:

1. A *véletlen komponens*: az Y komponensei független, normális eloszlású, σ^2 állandó varianciájú változók, amelyekre igaz: $E(Y) = \mu$;

2. A *szisztematikus komponens*: az $x_1, \dots, x_p$ változók egy η lineáris prediktort alkotnak

$$\eta = \sum_{1}^{p} x_j \beta_j;$$

   formában.

3. Az *összekötő komponens*: az összeköti a véletlen, és a szisztematikus komponenst:

$$\mu = \eta.$$

Ez az általánosítás bevezet

- egy újabb szimbólumot, az η-t, amellyel a lineáris prediktort jelöljük,
- és egy haramadik komponenst, amely a másik két komponenst köti össze, jelen esetben identikus módon. Azonban ha

$$\eta_i = g(\mu_i),$$

akkor a $g(.)$ függvényt linkfüggvénynek (összekötő függvénynek) nevezzük.

Ebben az összefüggésben a klasszikus lineáris modell esetén az első komponensnél normális eloszlással dolgozunk, a harmadik komponens pedig identikus függvény. A GLM-ek két kiterjesztést is megengednek.

- Az első komponens nem csak normális eloszlású lehet.
- A harmadik komponensbeli link függvény bármilyen monoton, differenciálható függvény lehet.

Először a kibővített első komponenst tárgyaljuk.

4.3.2. Likelihood függvények a GLM-hez

Tegyük fel, hogy az Y komponensei az exponenciális családból származnak, így

$$f(y;\theta;\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\theta)} + c(y;\theta)\right\}, \tag{4.5}$$

ahol $a(.)$, $b(.)$, $c(.)$ függvények adottak. Ha ϕ ismert, akkor ez egy exponenciális családba tartozó modell, θ kanonikus paraméterrel. Ha ϕ ismeretlen, akkor lehetséges, hogy nem kétparaméteres exponenciális családba tartozik. Így a normális eloszlás esetén

$$f_Y(y;\theta;\phi) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{\frac{-(y-\mu)^2}{2\sigma^2}\right\} = \exp\left\{\frac{y\mu - \frac{\mu^2}{2}}{\sigma^2} - \frac{1}{2}\left(\frac{y^2}{\sigma^2} + log(2\pi\sigma^2)\right)\right\},$$

így $\theta = \mu$, $\phi = \sigma^2$, és

$$a(\phi) = \phi, \qquad b(\theta) = \frac{\theta^2}{2}, \qquad c(y;\phi) = -\frac{1}{2}\left\{\frac{y^2}{\sigma^2} + log(2\pi\sigma^2)\right\}.$$

Az $l(\theta;\phi;y)$ helyére az $f_Y(y;\theta;\phi)$ log-likelihood függvényt írjuk (mivel $l(\theta;\phi;y) = f_Y(y;\theta;\phi)$), amely θ függvénye, és amelyhez ϕ, y paraméterek adottak. Az Y átlaga és varianciája könnyen megkapható az alábbi ismert összefüggések alapján:

$$E\left(\frac{\partial l}{\partial \theta}\right) = 0 \tag{4.6}$$

és

$$E\left(\frac{\partial^2 l}{\partial \theta^2}\right) + E\left(\frac{\partial l}{\partial \theta}\right)^2 = 0. \tag{4.7}$$

A (4.5) összefüggésből

$$l(\theta;y) = \frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi),$$

így

$$\frac{\partial l}{\partial \theta} = \frac{y - b'(\theta)}{a(\phi)} \tag{4.8}$$

és

$$\frac{\partial^2 l}{\partial \theta^2} = \frac{-b''(\theta)}{a(\phi)}, \tag{4.9}$$

ahol θ szerint deriváltunk.

A (4.6), és a (4.8) egyenletekből kapható a

$$0 = E\left(\frac{\partial l}{\partial \theta}\right) = \frac{\mu - b'(\theta)}{a(\phi)},$$

így

$$E(Y) = \mu = b'(\theta).$$

Hasonlóan, a (4.7), és a (4.9) egyenletekből:

$$0 = -\frac{b''(\theta)}{a(\phi)} + \frac{var(Y)}{a^2(\phi)},$$

ebből következik, hogy

$$var(Y) = b''(\theta)a(\phi).$$

Így, az Y varianciája két függvényből származik,

- míg az egyik $b''(\theta)$, csak a kanonikus paramétertől függ (és emiatt az átlagtól), és varianciafüggvénynek nevezzük,
- addig a másik $a(\phi)$,θ-tól független, és csak a ϕ-től függ.

A varianciafüggvény μ-nek a függvénye, ezért $V(\mu)$-vel szokás jelölni. Az $a(\phi)$ függvényt általában az alábbi formában írják le:

$$a(\phi) = \frac{\phi}{w},$$

ahol ϕ konstans az összes megfigyelésre (szokás σ^2-el jelölni, és diszperziós paraméternek nevezni), és w ismert súly minden megyfigylésre. Így normál modell esetén m átlagú független megfigyelésekkel a képlet

$$a(\phi) = \frac{\sigma^2}{m}$$

formában írandó, ahol $w = m$.

A (4.5) összefüggés elemeit az egyes eloszlásokra a (4.1) tartalmazza.

4.3.3. Link függvények

A linkfüggvények összekötik az η lineáris prediktort az y adat μ értékével. Klasszikus lineáris modellek esetén a lineáris prediktor identikus függvény, és az a kézenfekvő, hogy a linkfüggvény is identikus legyen, hiszen mind a μ, mind az η a valós számsík bármely értékét felveheti.

Ugyanakkor, ha Poisson eloszlású adatokkal dolgozunk, akkor a $\mu > 0$ feltételnek teljesülnie kell, emiatt az identikus linkfüggvény már nem megfelelő, részben amiatt, mert az η felvehet negatív értékeket, ám a μ nem. Ilyen esetben a log link függvény használatos, a $\eta = \log \mu$, és annak inverze, a $\mu = e^{\eta}$.

	Normális	Poisson	Binomiális
Jelölés	$N(\mu, \sigma^2)$	$P(\mu)$	$\frac{B(m,\pi)}{m}$
y korlátai	$(-\infty; \infty)$	$(0; \infty)$	$\frac{(0;m)}{m}$
Diszperziós paraméter: ϕ	$\phi = \sigma^2$	1	$\frac{1}{m}$
Kummulált függvény: $b(\theta)$	$\frac{\theta^2}{2}$	$\exp(\theta)$	$\log(1 + e^\theta)$
$c(y; \phi)$	$-\frac{1}{2}\left(\frac{y^2}{\phi} + \log(2\pi\phi)\right)$	$-\log y!$	$\log\begin{pmatrix} m \\ my \end{pmatrix}$
$\mu(\theta) = E(Y; \theta)$	θ	$\exp(\theta)$	$\frac{e^\theta}{1+e^\theta}$
Kanonikus link: $\theta(\mu)$	identitásfv.	log	logit
Variancia függvény: $V(\mu)$	1	μ	$\mu(1-\mu)$

	Gamma	Inverz
Jelölés	$G(\mu, \nu)$	$IG(\mu, \sigma^2)$
y korlátai	$(0; \infty)$	$(0; \infty)$
Diszperziós paraméter: ϕ	$\phi = \nu^{-1}$	$\phi = \sigma^2$
Kummulált függvény: $b(\theta)$	$-\log(-\theta)$	$-(-2\theta)^{\frac{1}{2}}$
$c(y; \phi)$	$\nu\log(\nu y) - \log y$ $-\log\Gamma(\nu)$	$-\frac{1}{2}\left\{\log(2\pi\phi y^3) + \frac{1}{\phi y}\right\}$
$\mu(\theta) = E(Y; \theta)$	$-\frac{1}{\theta}$	$(-2\theta - 2)^{-\frac{1}{2}}$
Kanonikus link: $\theta(\mu)$	reciprokfv.	$\frac{1}{\mu^2}$
Variancia függvény: $V(\mu)$	μ^2	μ^3

4.1. táblázat. A (4.5) összefüggésének elemei különböző eloszlásokra

Ha binomiális eloszlásból származó adatokkal foglalkozunk, akkor a $0 < \mu < 1$ feltételnek teljesülnie kell, ekkor három lehetséges link függvény is létezik:

1. logit
$$\eta = \log\left\{\frac{\mu}{1-\mu}\right\};$$

2. probit
$$\eta = \Phi^{-1}(\mu);$$
ahol $\Phi(\cdot)$ a normális kumulatív eloszlásfüggvény.

3. kiegészítő log-log
$$\eta = \log\left\{-\log(1-\mu)\right\}.$$

A hatvány linkfüggvények fontosak pl. a pozitív átlagú megfigyelések esetén. A linkfüggvények ezen típusa leírható
$$\eta = \frac{\mu^\lambda - 1}{\lambda},$$
az alábbi korlátozó feltételekkel
$$\eta = \log\mu; \qquad \text{amennyiben} \qquad \lambda \to 0,$$
vagy
$$\eta = \begin{cases} \mu^\lambda; & \lambda \neq 0, \\ \log\mu; & \lambda = 0. \end{cases}$$
Az első forma egyenletes átmenetet biztosít, ahogy lambda 0-ba konvergál, de mindkét esetben külön kezelni kell a $\lambda = 0$ esetet.

4.3.4. Elégséges statisztika és kanonikus linkek

Egy statisztika elégséges, ha nem létezik más olyan statisztika, amelyet ugyanabból a mintából generálunk, és több információt tartalmaz. [13]

A (4.1) táblázatokban lévő minden eloszláshoz tartozik egy különleges link függvény, amelyhez létezik egy elégséges statisztika, amely dimenzióban megegyezik β-val a $\eta = \sum x_j\beta_j$ lineáris prediktorban. Ezeket kanonikus link függvényeknek nevezzük, és akkor fordul elő, ha

$$\theta = \eta,$$

ahol θ a kanonikus paraméter, amit a (4.5)-ban definiáltunk, és megmutattunk a (4.3.2) fejezetben található táblázatban. Az eloszlásokhoz tartozó kanonikus linkfüggvények a következők:

normális $\eta = \mu$,

Poisson $\eta = \log \mu$,

binomiális $\eta = \log\left\{\frac{\pi}{1-\pi}\right\}$,

gamma $\eta = \mu^{-1}$,

inverz $\eta = \mu^{-2}$.

A kanonikus linkek számára a elégséges statisztika X^TY az alábbi komponensekkel

$$\sum_i x_{ij}Y_i, \qquad j = 1, \ldots, p,$$

az összes elemet szummázva. A kanonikus linkek alkalmazás kielégítő statisztikai tulajdonságokhoz vezet, ám vannak olyan eloszlású minták, ahol ezek nem alkalmazhatóak. Ezen esetek ismertetésétől eltekintünk, a dolgozat további részeihez nincs szükségünk erre az információra.

4.4. A becslés jósága

A becslés jósága több módon is mérhető, az általánosított lineáris módszer a legkisebb négyzetek módszer egy változatát használja, amely a becsült, és az eredeti, adatban szereplő pontok távolságát adja meg. Jelölése:

$$D(y;\hat{\mu}).$$

A (4.3.2) fejezetben található táblázatokhoz az alábbi hibafüggvények tartoznak:

normális $\sum(y-\hat{\mu})^2$,

Poisson $2\sum\left\{y\log\left(\frac{y}{\hat{\mu}}\right)-(y-\hat{\mu})\right\}$,

binomiális $2\sum\left\{y\log\left(\frac{y}{\hat{\mu}}\right)+(m-y)\log\left[\frac{m-y}{m-\hat{\mu}}\right]\right\}$,

gamma $2\sum\left\{-\log\left(\frac{y}{\hat{\mu}}\right)+\frac{y-\hat{\mu}}{\hat{\mu}}\right\}$,

inverz $\sum\frac{(y-\hat{\mu})^2}{\hat{\mu^2}y}$.

A módszer ezek után egy optimalizálási feladat, melynek sok megoldási lehetősége van. A módszert sok statisztikai programba beépítették, sok rendszerbe, többek között Octave-ba (6.1. fejezet) is implementálták.

4.5. Összegzés

A fejezet során bemutattam a Generalized Linear Model módszerét, amely sok feladatra felhasználható eredményesen felhasználható. A felhasznált irodalmak között található olyan, amelyben a főkomponens-analízist terjesztették ki exponenciális családokra[1]. A továbbiakban a szinguláris felbontást mutatom be, majd rátérek a G^2L^2M tárgyalására, mely részben a fejezetben tárgyalt módszert használja fel.

4.6. Bibliográfiai megjegyzések a fejezethez

A fejezet az alábbi könyvek megadott részeinek a fordítása, átfogalmazása, értelmezése.

[7] M. Collins, S. Dasgupta, és R. E. Schapire. *A generalization of principal components analysis to the exponential family.* T. G. Dietterich, S. Becker, and Z. Ghahramani (szerk.). Advances in Neural Information Processing Systems, volume 14, Cambridge, MA, 2002. MIT Press.

[11] P. McCullagh, és J.A. Nelder. *Generalized Linear Models.* 1-2. fejezet. CRC Press, 2nd edition, 1990.

[12] Sir Harold Jeffreys. *Theory of probability.* x. oldal. Oxford University Press, 3rd edition, 1998.

[13] Fisher, Ronald. *On the Mathematical Foundations Of Theoretical Statistics.* Phil. Trans. R. Soc. Lond. 1922.

[1] [7] irodalom foglalkozik ezzel a témával

5. fejezet

A G^2L^2M

A fejezetben bemutatok egy olyan statisztikai becslő eljárást, mely a nemlineáris regresszió, és a főkomponens-analízis tulajdonságait kombinálja. A Generalized^2 Linear^2 Model egy X mátrixot bont fel egy egyszerűbb $f(g(A)h(B))$ alakra, ahol A és B alacsony rangú mátrixok, f, g és h pedig link függvények. Az Általánosított^{2}Lineáris^2 Modell speciális esetei között hasznos modellek találhatók meg, úgy mint főkomponens-analízis (PCA), exponenciális családra kiterjesztett főkomponens-analízis (ePCA), Lineáris regresszió, Általánosított Lineáris Modell (GLM).

A fejezet során bemutatom a modellt, egy iteratív módszert, mely optimalizálja a $(GL)^2M$ paramétereit, és közismert algoritmusokra redukálja a feladatot. [14]

5.1. A módszer bevezetése

Legyen X $m \times n$ méretű, független, ismeretlen eloszlású adatokat tartalmazó adatmátrix. X minden oszlopa egy-egy megfigyelést (tanító példát), és minden sora egy-egy tulajdonságot (attribútumot) tartalmaz. Gyakran értelmes megközelítés, ha feltételezzük, hogy az adatok redundánsak, azaz létezik egy olyan l amely X minden vagy majdnem minden információját tartalmazza, viszont kevesebb attribútumot tartalmaz.

Ha a csökkentett attribútumok lineáris függvényei az eredeti tulajdonságoknak, és X elemei normális eloszlást követnek, akkor a csökkentés azt jelenti, hogy az X mátrixot helyettesíthetjük U és V mátrixok szorzatával, kis négyzetes hibával. Ez gyakorlatilag a szinguláris felbontás, ahol U a baloldal altér, V^Ta jobboldali altér első l dimenzióját tartalmazza. Mivel ez nem határozza meg egyértelműen az U és V mátrixokat, a szinguláris felbontás további megszorításokat alkalmaz.

Az SVD-t számtalan alkalmazásban sikeresen alkalmazták, többek közt az eigenfaces arcfelismerő rendszerben, és az adattömörítés terén is, de két olyan megszorítást tartalmaz, amely nagyban korlátozza a felhasználhatóságát:

1. A hiba számítására a négyzetes hibafüggvényt használja, amely azt jelenti, hogy az 1000 és az 1010 közötti eltérés ugyanakkorának számít, mint a 3 és -7 közötti.

2. Az X mátrix és az U, V mátrixok között lineáris kapcsolat van.

Ehelyett az alábbi modellt javasolja Gordon ([14]):

$$\hat{X} = f(g(A)h(B)) \tag{5.1}$$

X számított értéke. Továbbá azt is, hogy ne csak kvadratikus loss-függvényeket használjanak a hibák értékének kiszámításakor, sem az $(X - \hat{X})$, sem az A illetve B paramétermátrix esetén. Az

$$f: \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n} \quad g: \mathbb{R}^{m\times l} \to \mathbb{R}^{m\times l} \quad h: \mathbb{R}^{l\times n} \to \mathbb{R}^{l\times n}$$

fix függvényeket *link függvény*eknek hívják. A generalized linear models (4. fejezet) mintájára az 5.1 egyenletet G^2L^2M-nek hívjuk, mert:

- Generalized^2, mert link függvényeket használ mind az A és B paramétermátrix elemeihez, mind a $\hat{X}$ becsléséhez, és
- Linear^2, mert mint az SVD (3.2 .rész), ez a módszer is bilineáris.

Amennyiben egymáshoz tartozó link és loss függvényeket választunk, létezik egy olyan hatékony algoritmus, mely segítségével megkaphatjuk A és B értékét, X, f, g, h ismeretében. [14]

5.2. Összetartozó link és veszteség függvények

Minden alkalommal, amikor egy nemlineáris modell becslését próbáljuk optimalizálni, ügyelnünk kell, hogy ne szoruljunk be egy lokális minimumpontba. Erre egy jó példa, ha egy $\theta \in \mathbb{R}^d$ paraméterű szigmoid függvényt próbálunk illeszteni $x_i \in \mathbb{R}^d$ tanuló adatokra, és $y_i \in \mathbb{R}^d$ kívánt célértékekre, négyzetes hibafüggvényt (squared loss) használva:

$$L = \sum_i (y_i - \hat{y}_i)^2 \qquad \hat{y}_i = \text{logit}(z_i) \qquad z_i = x_i \cdot \theta \qquad \text{logit}(z) = (1 + e^{-z})^{-1}.$$

Még kis tanuló mintára is L lokális minimumértékeinek száma exponenciálisan nő a d dimenzióval. Ellenben, ha ugyanazt az $\hat{y}_i$ becslést optimalizáljuk $\sum_i [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$ logaritmikus hibafüggvénnyel[1] (négyzetes hibafüggvény helyett), konvex optimalizációs problémát kapunk. Mivel a logisztikus linkfüggvény a log hibafüggvénnyel konvex optimalizációs problémát alkot, ezért azt mondjuk, hogy ezek összetartozóak. Az összetartozó link-loss párok fontosak, mert egy konvex probléma megoldása sokkal egyszerűbb, és ezáltal gyorsabb, mint egy nemkonvex problémáé.

Bármely $F(z)$ konvex függvényt használhatjuk, hogy összetartozó link-loss párokat alkossunk. Az F-hez tartozó hibafüggvény:

$$D_F(z|y) = \sum_i [F(z_i) - y_i z_i + F^*(y_i)], \tag{5.2}$$

ahol $F^*(y)$ definiált, így $\min D_F(z|y) = 0$. (F^* F konvex duálisa, és D_F a z és y közötti általánosított Bregman távolság.)

(5.2) egyenlet nemnegatív, és globálisan konvex minden z_i-re, így θ-ra is (mivel z_i lineáris függvénye θ-nak). Ha F gradiense f, (5.2) deriváltja z_i szerint $f(z_i) - y_i$. Így (5.2) értéke akkor lesz 0, ha $y_i = f(z_i)$ minden i esetén, más szóval (5.2) lossfüggvényt használva y_i legjobb becslése $\hat{y}_i = f(z_i)$ lesz, így f a megfelelő linkfüggvény.

Két követelményt támasztunk a konvex duálisok irányában,

1. F^* mindig konvex, és
2. F^* gradiense legyen egyenlő f^{-1}-el. [14]

[1] log loss

5.3. (GL)^{2}M-hez tartozó veszteség függvények

A loss függvények kifejezetten fontosak a (GL)^{2}M-ek esetében, mivel három külön nemlineáris függvénynyel kell megbírkózni. Általában nem tudjuk elkerülni a lokális minimumot, ehelyett az összesített veszteségfüggvényt tesszük konvexszá néhány paraméterre úgy, hogy a további paramétereket konstansnak vesszük.

A (GL)^{2}M-eket három link-, és hozzá tartozó loss függvény segítségével írjuk le. A különálló loss függvényekből előállítható egy összesített loss függvény (az előállítás a 5.4 részben szerepel). A későbbiekben a (GL)^{2}M illesztése ennek az összesített loss függvénynek a minimalizására (az adott paramétereket minimalizálva) redukálható. A különböző link függvények választása különböző modelleket, ennélfogva X különféle felbontását is eredményezi.

A link függvények kiválasztása az a pont, ahol kihasználhatjuk a tudásunk

- az X mátrixon található zajról, és
- hogy mely A és B paramétermátrixok a priori valószínűsége a legnagyobb.

A használható link függvények közé tartoznak:

- $f(x) = x$ - négyzetes hiba, normál eloszlású zaj
- $f(x) = \log x$ - normalizálatlan KL-eltérés[2], Poisson eloszlású zaj
- $f(x) = (1 + e^{-x})^{-1}$ - log-loss, Bernoulli eloszlású zaj.

A loss függvények csupán az analízishez szükségesek, az összes algoritmus a link függvényekre és (néhány esetben) azok deriváltjaira támaszkodik. Így kiválaszthatjuk a veszteségfüggvényt, és differenciálhatjuk, hogy megkapjuk a (loss függvényhez tartozó) megfelelő linkfüggvényeket, vagy kiválasztunk egy megfelelő linkfüggvényt, és nem törődünk a loss függvénnyel. Annak érdekében, hogy analizálni tudjuk modellünket, a link függvényeknek valamely (valószínűleg ismeretlen) konvex függvény deriváltjának kell lennie.

A loss függvényeink $D_F,\ D_G,\ D_H$ lesznek, ahol

$$F : \mathbb{R}^{m\times n} \to \mathbb{R} \qquad G : \mathbb{R}^{m\times l} \to \mathbb{R} \qquad H : \mathbb{R}^{l\times n} \to \mathbb{R}$$

konvex függvények. A jelölést rugalmasan kezeljük, ezért F-et, G-t, és H-t is loss függvénynek fogjuk nevezeni;

F-et a becslés veszteségfüggvényének, deriváltját f-et a becslés link függvényének nevezzük, mely az X mátrixon mérhető zajt modellezi,

G és H, paraméter veszteségek függvényei, deriváltjaik g-t és h-t a paraméter link függvényének nevezzük, melyek A és B paraméterhalmaz legnagyobb a priori valószínűségű értékeit mutatja meg. Mivel F egy $m \times n$ méretű mátrixot kap bemenetként, azt mondjuk, hogy f ki-, és bemenete szintén $m \times n$ méretű mátrix lesz (hasonlóan g-nek és h-nak). [14]

5.4. A modell és egyenletei

Definiálunk egy (GL)^{2}M-et egy összesített loss függvény segítségével, amely az A és B paramétermátrixokat veti össze az X adatmátrixszal. Ha $U = g(A)$ és $V = h(B)$ helyettesítéseket elvégezzük, a

[2] Kullback-Leibler divergencia, két valószínűségi változó (P, Q) közötti távolság. Számítása: $D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$ [15]

(GL)^{2}M összesített loss függvény

$$L(U,V) = F(UV) - X \circ UV + G^*(U) + H^*(V) \tag{5.3}$$

lesz, ahol $A \circ B$ a „mátrix pont produktum", és gyakran $\mathrm{tr}(A^T B)$[3] formában írják.

A (5.3) kifejezés három Bregman távolság összege, figyelmen kívül hagyva azokat a kifejezéseket, melyek U-tól és V-től függetlenek, azaz $D_F(UV|X) + D_G(0|U) + D_H(0|V)$. Az F eltérés függvény az UV-t X felé tolja, míg a G és a H kis U és V értékeket részesít előnyben.

Hogy a későbbiekben bizonyíthassuk (5.3)-t, megnézhetjük, hogy mi történik, ha U és V szerint deriváljuk. Ekkor az alábbi két egyenletet kapjuk:

$$U^T(X - f(UV)) = B \tag{5.4}$$

$$(X - f(UV))V^T = A \tag{5.5}$$

A két egyenlet értelmezéséhez két példát hozunk.

1. Ha G^* értékét 0-ra állítjuk (így nem szükséges U-t és V-t kis értékűre állítani), (5.4)

$$U^T(X - f(UV)) = 0 \tag{5.6}$$

   egyenletté írható át, ami azt jelenti, hogy a hibamátrix minden oszlopa ortogonális U minden oszlopára.

2. Ha a becslő link függvényt $f(UV) = UV$-nak választjuk, (??)

$$U^T UV = U^T X$$

   alakban írható fel, amely szerint adott U mellé kell V-t választani úgy, hogy a négyzetes eltérésük X-től minimális legyen. [14]

5.5. Algoritmusok a (GL)^{2}M paramétereinek illesztésére

A (5.4, 5.5) egyenleteket több algoritmus segítségével is meg lehet oldani. Például a gradient descend (csökkenő gradiens - GD) módszerrel U-n és V-n, vagy A-n és B-n alkalmazva. Használhatjuk még a generalized gradient descent (általánosított csökkenő grádiens - GGD) módszert is (α tanulási együtthatóval):

$$A \leftarrow_\alpha (X - f(UV))V^T, \tag{5.7}$$

$$B \leftarrow_\alpha U^T(X - f(UV)). \tag{5.8}$$

Ezen algoritmusok előnye, hogy nincs szükség további megszorításokra F, G, és H függvényekre. [14]

Dolgozatomban ez utóbbi módszer implementálását oldottam meg.

[3] A trace függvény egy mátrix főátlójában található értékek összegét adja meg, azaz $\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}$

5.6. Összegzés

A fejezet során bemutatásra került a nemlineáris regresszió és faktoranalízis modellek egy olyan általánosítása, mely kiválthatja a PCA-t (3.1), és az SVD-t (3.2) is. A továbbiakban a GNU Octave rendszer, és az implementáció leírása következik.

5.7. Bibliográfiai megjegyzések a fejezethez

[14] Geoffrey G. Gordon. *Generalized^2 Linear^2 Models.* Advances in Neural Information Processing Systems, volume 15, Cambridge, MA, 2003. MIT Press.

[15] Kullback, S.; Leibler, R.A. *On Information and Sufficiency.* The Annals of Mathematical Statistics 22 (1): 79–86. 1951.

6. fejezet

Az algoritmus implementálása

A fejezet során bemutatom a rendszert, melyre az algoritmust írtam, majd bemutatok egy ismert optimalizációs módszert, amelyet implementáltam az algoritmus során. Majd magát a $(GL)^2M$ implementálását mutatom be, és egy példát is végrehajtok segítségével.

6.1. Az Octave

Ebben a részben azt a GNU Octave rendszert ismertetem, melyet munkám során használtam. Kipróbáltam mind a Linux, mind a Windows verziót, de az utóbbi nem más, mint a program Linux verziója, csupán megfelelő környezetet biztosít számára a Windows alatt. Nyílvánvaló, hogy az említett tulajdonság miatt lassabb lesz Linuxos társánál, ezt mérlegelve, a Linux verziót használtam diplomamunkám elkészítése közben.

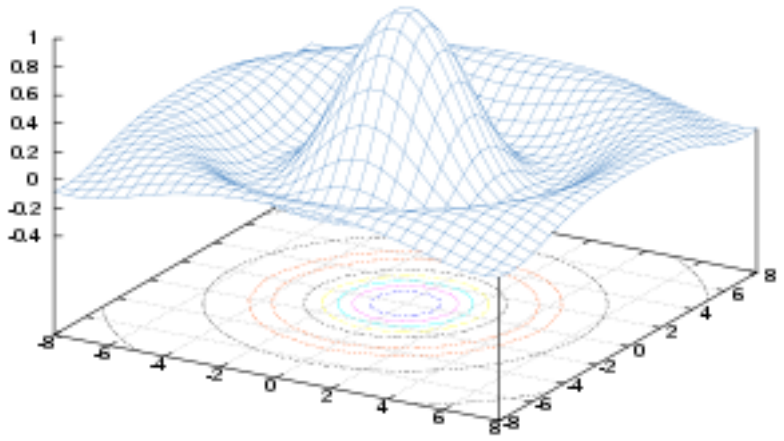
A GNU Octave egy olyan nyílt forráskódú (GNU GPL licenszű[1], tehát szabadon felhasználható, módosítható, terjeszthető), magasszintű programozási nyelv, amely elsősorban numerikus műveletek végzésére használatos. Kényelmes parancssori felületet biztosít lineáris, és nemlineáris problémák numerikus megoldásához, valamint egyéb numerikus kísérletekhez, egy olyan nyelven, amely (majdnem teljesen) kompabilitis a MatLab rendszerrel.

Az Octave rendszerhez sok kiegészítő készült, melyek segítségével ma már egy-egy parancs kiadásával lehet:

- egyszerű, lineáris algebrai problémákat megoldani,
- nemlineáris egyenletek gyökeit megkeresni,
- közönséges függvényeket integrálni,
- polinomokat manipulálni,
- és még sok más matematikai problémát megoldani.

Az Octave egyszerűen bővíthető és testreszabható a saját nyelvén íródott függvények segítségével, vagy valamely más programnyelven (C, C++, ForTran) írt csomag kezelésével. [17]

[1] http://www.gnu.hu/gpl.html



6.1. ábra. Az Octave-ban kiadott 3 dimenziós plot parancs eredménye, a nem hivatalos Octave logó.

6.1.1. Történeti áttekintés

Az Octave rendszert eredetileg (1988-ban körül), egy egyetemi segédlet mellé írták, mint segédeszközt. A Wisconsin-i Madison egyetem oktatója, James B. Rawlings, valamint a Texasi egyetem oktatója, John G. Ekerdt munkáját eredetileg egy specializált feladat (kémiai reaktorok tervezési feladata) megoldására találta ki, de később úgy látta a szerzőpáros, hogy ez nagy megszorítás a program felhasználhatóságát tekintve, ezért flexibilisebb rendszer tervezésébe fogtak.

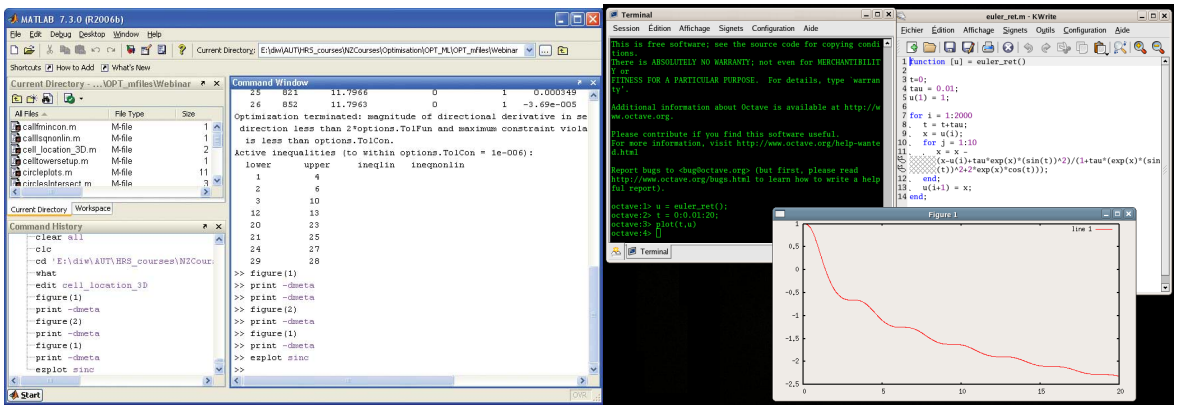
Sokan azt javasolták nekik, hogy használjanak inkább Fortran nyelvet az órájukon, ám amikor ezt a megoldást választották, a hallgatóknak túl sok idejükbe került, hogy megtalálják a hibát a Fortran kódjukban, így nem tudtak az igazi feladatra, a kémiai tervezésre koncentrálni. Ezért a tervezők úgy gondolták, hogy egy olyan interaktív rendszert, mint az Octave, a hallgatók többsége gyorsan megismeri az alapokat, és néhány óra elteltével magabiztosan használjhatja.

A valódi fejlesztést John W. Eaton kezdte meg 1992-ben, és 1993-ra már készen állt az első verzó, majd 1 évre rá, 1994-ben készült el a program 1.0-ás változata. Azóta sok fejlesztésen esett át, a 3.0-ás verzó 2007-ben látott napvilágot. [17]

6.1.2. Technikai adatok

Az Octave rendszert C++ programnyelven írták, STL könyvtárak felhasználásával. Az Octave parancsai futtatásához egy értelmezőt használ. A rendszert könnyű bővíteni különböző modulokkal, ezeket a munkamenet elején be kell tölteni, hogy használni lehessen függvényeit.

A rendszer a grafikonok, grafikák, diagrammok megjelenítéséhez a gnuplot programot használja.[18]



6.2. ábra. A Matlab (bal oldalon) és az Octave (jobb oldalon) munka közben

6.1.3. Octave, mint nyelv

Az Octave egy értelmező struktúrált programozási nyelv (hasonlóan a C nyelvhez), és sok C könyvtárat támogat. Az Octave programok függvényhívások sorozata, vagy szkript. A szitakszis a mátrixokon alapul, és az ezeken végzett különböző műveletek nagy számban implemetálva van a rendszerbe. Nem objektum orientált nyelv, viszont adatstruktúrák létrehozását támogatja.

Az Octave rendszerben implementáltak néhány kényelmi funkciót is, pl:

- Parancsok automatikus kiegészítése,
- Parancselőzmények tárolása,
- Tetszőleges hosszúságú függvény- argumentum, és visszatérési érték[2] használata. [18]

Kapcsolata a MatLab-bal. A rendszer (nagymértékben) kompatibilis a matlabban írt programokkal, olyannyira, hogy egyes függvényei olyan változókat is tartalmaznak az argumentumlistájukban, melyek nem szükségesek a működésükhöz. Ezeket megadva az Octave-ra írt programok futtathatóak MatLabban is.

Az alábbi tulajdonságok közösek a MatLabbal:

- Mátrixok, mint alapvető adattípus,
- Komplex számok kezelése,
- Beépített függvények, bővíthetőség,
- Felhasználók által bővíthető függvénytár. [18]

[2] Az Octave képes előre nem deklarált, tetszőleges mennyiségű argumentum kezelésére, és visszatérési érték adására.

Algorithm 6.1 A két paramétermátrix GGD algoritmusa

(5.7) alapján:

```
Ciklus, amíg a hiba > a pontosság
```
$A_{uj} = A_{regi} - \alpha * (X - f(g(A_{regi})h(B_{regi})) * h(B_{regi})^T$
$A_{regi} = A_{uj}$
```
Ciklus vége.
```

(5.8) alapján:

```
Ciklus, amíg a hiba > a pontosság
```
$B_{uj} = B_{regi} - \alpha * g(A_{regi})^T * (X - f(g(A_{regi})h(B_{regi}))$
$B_{regi} = B_{uj}$
```
Ciklus vége.
```

6.2. Az algoritmus

A diplomamunka írása során az eljárás változott a legtöbbet. Sok módszert kipróbáltam, míg egy olyan eljáráshoz nem jutottam, amely egy ésszerű kompromisszum.

A feladat leegyszerűsítve egy olyan optimalizációs eljárás, mely során a globális minimumát keressük egy függvénynek, jelen esetben a hiba, vagy más néven veszteség függvénynek (5.3. egyenlet).

Sokat kerestem különböző optimalizáló eljárások között, mert kézenfekvő megoldás, hogy egy már jól megírt, és sokak által alkalmazott kódot használjak. Az algoritmusok egy függvényt optimalizálnak, amely számomra tökéletes volt, hiszen ez volt a feladatom. A problémát az jelentette minden esetben, hogy a hibafüggvény visszatérési értékének mindenféleképpen egy számnak kell lennie. Mivel a $(GL)^2M$ a hibát a mátrix tagjai között értelmezi, így ez a megoldás nem lehetett jó; az összesített hiba visszaadása néhány iterációs lépésen belül túlcsordulást okozott az értékeken. Ez alapján világossá vált számomra, hogy nekem kell implementálnom az optimalizációs eljárást, hogy a hibákat megfelelően tudjam kezelni.

6.2.1. Az optimalizációs eljárás

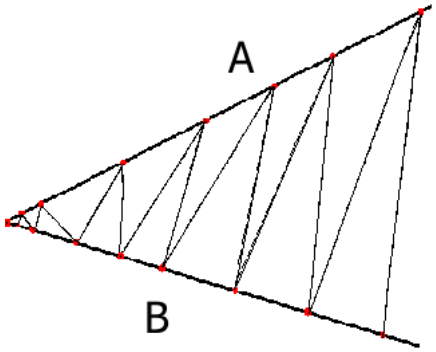
A feladat az (5.3) egyenlet lokális minimumának keresése, és a legtöbb algoritmus megköveteli ennek deriváltját is. Mivel 2 paramétermátrixunk is van, ezért két derivált fog keletkezni, (5.4) és (5.5).

A módszer megoldását GGD algoritmussal végeztem el, melyhez szükség van egy hibafüggvényre, valamint annak deriváltjaira is. Így (5.4) és (5.5) alapján felírhatjuk (5.8)-t, és (5.7)-t, melyeket ezután felhasználhatunk, és a (6.1) algoritmust fogalmazhatjuk meg.

A feladat során mindkét paramétermátrix értékeit illeszteni kell a megfelelő eredmény érdekében, ezért egyszer az egyik mátrix értékeit vesszük konstansnak, máskor a másikét. Ezáltal egy olyan iterációs folyamat jön létre, mely a becsülendő értékek felé konvergál. Hogy szemléletes legyen, a (6.3) ábrán látható a leírt iteráció sematikus ábrája.

Az algoritmust egészen addig kell futtatni, míg a hiba kisebb nem lesz egy előre kiválasztott nagyon kicsi értéknél. Ezt a konvergencia feltételének is szokták nevezni.

Az optimalizációs eljárást a ggllm függvény hajtja végre.



6.3. ábra. Iterációs lépések sematikus ábrája, ahol a két vastag vonal jelképezi A, ill. B paraméterhalmaz hibáját.

6.2.2. A ggllm eljárás, és annak részei

A ggllm függvényhez több segédfüggvény is tartozik, először ezeket tekintem át, majd rátérek a főfüggvény leírására.

Az alkalmazásban található függvények:

- function [A,B] = ggllm (X, l, iter=500, alfa=0.01, prec=0.01, link="normal",plotting=0),
- function savedata(i, err),
- function ggllmtest(number),
- valamint a linkfüggények.

6.2.2.1. Link függvények

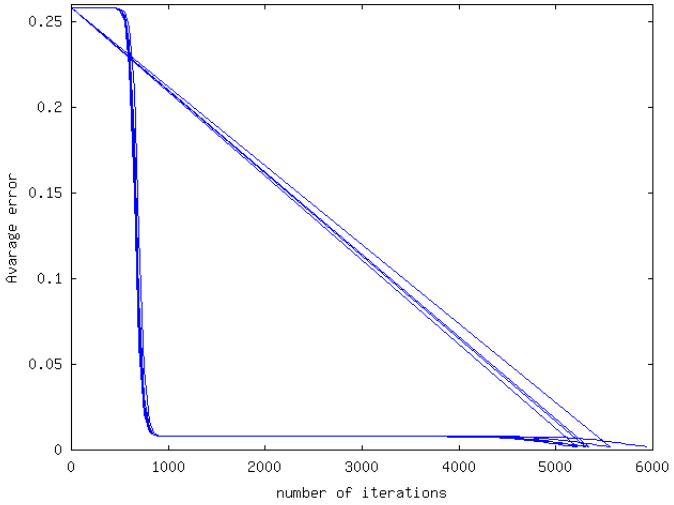
Ahhoz, hogy megfelelően működhessen az eljárás, szükség van kisegítő függvényekre, melyek a link függvényeket implementálják. Ezek felépítése egyszerű, a funkciójuk viszont annál fontosabb, hiszen ezeket a függvényeket hívja meg az eljárás minden alkalommal, amikor f, g, ill. h függvényekkel kell számolnia.

Összesen ötféle függvényt implementáltam, ezek a Normális, Poisson, Bernoulli, Gamma, illetve az Inverz zajokhoz megfelelő linkfüggvények. Az adott eljárások a (4.1) táblázatban található link függvények implementálása.

6.2.2.2. A savedata függvény

A savedata egy állományt nyit meg, és írja bele a felbontás során kapott adatokat. Elmenti az iteráció sorszámát, valamint az átlagos hiba értékét. Az állományt a munkakönyvtárba menti, ha nem létezik akkor létrehozza. Hibát nem vizsgál, tehát ha olyan könyvtárban dolgozunk, ahol nincs írásjogunk, nem fog működni az elemzési szempontból fontos eljárás[3].

[3] A merevlemezre írt adatok felhasználásával lehet a hiba alakulásáról grafikont készíteni. A (6.4) ábra is így készült.



6.4. ábra. A ggllmtest meghívásának eredménye

Algorithm 6.2 A ggllmtest meghívásának eredménye

```
octave:1> ggllm_GGDv02 # a függvényt tartalmazó fájl beolvasása
octave:2> ggllmtest    # a tesztfüggvény meghívása
Start                  # parancs kezdete
66                     # a metódushoz szükséges idő sec-ben
Loading...             # adatok beolvasása fájlból
Plotting...            # kirajzolás
```

6.2.2.3. A ggllmtest függvény

Ezt a függvényt azért implementáltam, hogy a témában ismeretlen felhasználó lásson egy példa függvényhívást, valamint hogy szemmel követhesse a hiba változását az iterációk függvényében. Az eljárást elláttam a szükséges dokumentációval, tehát a felhasználó angol nyelvű segítséget kap a

```
help ggllmtest
```

parancsra.

Ha a felhasználó nem adja meg a „number” változó értékét, tehát hogy hány alkalommal hajtódjon végre a dekompozíció, akkor ez alapértelmezetten 5 alkalommal történik meg.

A függvény futása során az alábbi információkkal látja el a felhasználót:

- a feladatok elvégzéséhez szükséges időről, valamint
- a hibáról készített grafikonnal (6.4. ábra, és 6.2. algoritmus).

6.2.2.4. A ggllm függvény

A függvényt több részre oszthatjuk az elvégzendő feladatok szempontjából.

1. A legelső rész az eljárás dokumentációja, mely a

   ```
   help ggllm
   ```

   parancsra a függvény leírását írja ki a képernyőre jelmagyarázattal és egy példával együtt.

2. A következő részben az argumentumok ellenőrzése következik, mely során vizsgálja, hogy a minimális paramétereket (X adatmátrix, l megtartani kívánt tulajdonság) megadta-e a felhasználó. Ennek hiányában a program hibaüzenettel tér vissza az Octave rendszerbe.

3. A harmadik részben kiválasztásra kerül a link függvény, és f, g, h függvényeknek átadásra kerül, így ezek használhatóvá válnak.

4. A továbbiakban A és B paramétermátrixok kezdeti értékét, valamint a konvergencia feltétel mátrixot generálja a program. Ez utóbbi egy olyan mátrix, melynek minden értéke a „prec" bemeneti értékkel egyezik meg. Ezt a GGD algoritmus során használja majd fel a program.

5. A GGD algoritmust, attól függően, hogy véges iterációt vagy időkorlátot választ a felhasználó, más-más feltételpárral hívja meg a program. A feltételpár egyik része az iteráció- ill. időtartamkorlátot ellenőrzi, a másik pedig azt, hogy a konvergencia feltételben megadott „prec" értéknél alacsonyabb-e a hiba az egyes tagokon. A GGD algoritmusának törzsét a (6.1). algoritmus írja le

6. A konvergálás után az eljárás feladata az eredmények kiírása.

Ahhoz, hogy a fent leírt részek működhessenek, a programnak az alábbi bemenetekre van szüksége:

X adatmátrixra

l értékre, mely közvetve azt mondja meg, hány dimenziót szeretnénk elhagyni

iter az iterációk számára, mely ha nagyobb, mint 0, az iterációk maximális számát jelenti, ellenkező esetben ilyen korlát nincs, csak egy maximális idő van megszabva (ez az idő könnyen módosítható egy változó segítségével a programban)

alpha tanulási tényezőre, melynek beállítása csak a felhasználótól függ, az alapértelmezett érték többszöri teszt eredménye

prec a precizitásra, vagy más néven a konvergencia korlátra, ha ennél kisebb az eltérés az eredeti mátrix értékeitől, a program úgy értelmezi, az értékek megfelelőek

link a link függvény típusára, melyet az X adatmátrixon feltételezett zaj alapján érdemes megválasztani

plot a naplózó mód ki/bekapcsolására használható. Amennyiben szeretnénk, hogy ne jelenjen meg szöveges kiértékelés a folyamat végén, és a hibák értékét 50 iterációnként naplózzák, 1-es értéket kell beállítanunk.

6.2.3. Példák

Ebben a részben néhány felbontást mutatok be, hogy prezentáljam a függvény működését.

A (3.2.4) részben már bemutattam egy példát, melynek bemenő adatait most a ggllm függvénynek adom bemenetként.

Példa 1 Mielőtt meghívnánk a függvényt, leírom még egyszer a feladatot, és a megoldás lépéseit:

- Először létrehozok egy $n \times m$ méretű mátrixot exponenciális eloszlású elemekkel, majd néhány oszlop értékét megnövelem normális eloszlásból származó zajjal.

  - Exponenciális eloszlású adatokból 500 dimenziós, 700 megfigyelést tartalmazó mátrix létrehozása:

    ```
    X = rande (700,500);
    ```

  - Zajmátrix létrehozása, az utolsó 200 dimenziót a hibával, a maradékot 0-val feltöltve:

    ```
    H=[zeros(700,300),randn(700,200)];
    ```

  - A mátrix létrehozása:

    ```
    M = X+H;
    ```

- Az eljárás elindítása előtt meghatározzuk, hogy mekkora legyen l értéke. Mivel itt tudjuk mekkora a zajjal terhelt rész, adott a 200-as érték.

- Következő lépésként azt kell eldöntenünk, hogy milyen zajjal terhelt az adathalmaz. Megintcsak könnyű a dolgunk, választhatjuk nyugodtan a normális eloszlást.

- Szerencsére ezek az értékek alapértelmezettek, így a függvény meghívható a következő módon:

  ```
  [A,B] = ggllm(M,200,0);
  ```

- Erre válaszképpen a következőt láthatjuk:

  ```
  converged in 125367 iteration in 240 seconds.
  Avarage of errors: 0.000548
  ```

  A megoldás közli a paramétermátrixokat is, ám ennek ismertetésétől itt eltekintek.

Példa 2 Egy másik egyszerű példa egy nevezetes mátrix vizsgálata. Erre a feladatra én a Hilbert mátrixot választottam; egy 5×5-ös méretű mátrixot bontsunk fel, l értékét 3-ra választva!

- Ezt egy lépésben is megtehetjük:

  ```
  [A,B] = ggllm(hilb(5),3,0);
  ```

  vagy ha inkább rögtön egy grafikont szeretnénk látni a hiba csökkenéséről, meghívhatjuk az előre implementált ggllmtest függvényt is. Fontos tudni, hogy az utóbbi esetben eredményként nem kapjuk meg a felbontás eredményeként keletkezett A és B paraméterhalmazokat!

  Ennek eredménye:

  ```
  converged in 6341 iteration in 10 seconds.
  Avarage of errors: 0.000216
  ```

6.3. Összegzés

A fejezet során bemutattam az Octave rendszert, majd magát az algoritmust. Az algoritmus még nem tökéletes a teljesítmény szempontjából, és nagy mátrixok esetén is problémákba ütközhet használata. Terveim között szerepel ennek továbbfejlesztése.

6.4. Bibliográfiai megjegyzések a fejezethez

[16] John W. Eaton. *GNU Octave Manual.* Network Theory Limited. 3rd edition, 2007.

[17] John W. Eaton. *About Octave.* http://www.gnu.org/software/octave/about.hml. 2009.04.20.

[18] Octave.org. *Frequently asked questions about Octave (with answers).* http://www.gnu.org/software/octave/FAQ.html#dir. 2009.04.20.

7. fejezet

Összegzés

Dolgozatom során bevezettem a Generalized^2 Linear^2 Models módszerét a becsléselméleti alapfogalmak, alapvető dimenziócsökkentő eljárások és a GLM előzetes leírásával. Ezután jellemeztem a felhasznált rendszert, és leírtam az implementált algoritmust.

Munkám eredményeképpen.

- Betekintést nyertem a dimenziócsökkentés kérdéskörébe, és megpróbáltam több aspektusból is megközelíteni azt.
- Egy számomra eddig ismeretlen programozási környezetet is megismertem, melyet úgy érzem sokszor tudok majd hasznosítani az elkövetkező munkáim során.
- A dolgozat megírásához a T_EX-et is megismertem, amit a későbbiekben is használni fogok.
- Implementáltam egy olyan dimenziócsökkentő eljárást, amely eddig nem volt elérhető.

További terveim között szerepel.

- az implementált program közkinccsé tétele, hogy azt bárki szabadon használhassa,
- a program tökéletesítése,
- más optimalizációs eljárások implementálása ugyanerre a problémára,
- ezek összehasonlítása a legeffektívebb eljárás megtalálása érdekében.

Remélem, hogy munkám gyümölcse eléri a megfelelő közösséget, és hasznos időt takarít meg azoknak, akik pont egy ilyen eljárást keresnek.

Irodalomjegyzék

[1] Fazekas István (szerk.). *Bevezetés a matematikai statisztikába.* II-III. fejezet. Kossuth Egyetemi Kiadó, 1997

[2] Fazekas István. *Valószínűségszámítás (Számítógépes segédlet).* IV. fejezet, 1. Debreceni Egyetem, 2000.

[3] MTA MMSz. *A maximum likelihood becslésről.* http://www.muszeroldal.hu/assistance/ml.pdf. 2009.04.19.

[4] Kabos Sándor. *Statisztika II. Becslés.* http://kabos.web.elte.hu/matstat/stat5e.pdf. 2009.04.19.

[5] Fazekas István (szerk.). *Bevezetés a matematikai statisztikába.* X.2. fejezet. Kossuth Egyetemi Kiadó, 1997.

[6] Jonathon Shlens. *A Tutorial on Principal Component Analysis.* http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf. 2009.04.19.

[7] M. Collins, S. Dasgupta, és R. E. Schapire. *A generalization of principal components analysis to the exponential family.* T. G. Dietterich, S. Becker, and Z. Ghahramani (szerk.). Advances in Neural Information Processing Systems, volume 14, Cambridge, MA, 2002. MIT Press.

[8] Rózsa Pál. *Lineáris Algebra és alkalmazásai.* Tankönyvkiadó, 3. kiadás, 1991.

[9] Bodon Ferenc. *Adatbányászati algoritmusok.* http://people.inf.elte.hu/ktk/Infkez1/adatbanyaszat.pdf. 2009.04.10.

[10] Hren Brunó. *Méréstechnikai és kemometriai módszerek fejlesztése a Fourier-transzformációs Infravörös és Raman-spektroszkópiában.* Doktori (PhD) értekezés, Veszprém 2008. http://twilight.vein.hu/phd_dolgozatok/hrenbruno/Hren_Bruno_Ertekezes.pdf. 2009.04.11.

[11] P. McCullagh, és J.A. Nelder. *Generalized Linear Models.* 1-2. fejezet. CRC Press, 2nd edition, 1990.

[12] Sir Harold Jeffreys. *Theory of probability.* x. oldal. Oxford University Press, 3rd edition, 1998.

[13] Fisher, Ronald. *On the Mathematical Foundations Of Theoretical Statistics.* Phil. Trans. R. Soc. Lond. 1922.

[14] Geoffrey G. Gordon. *Generalized^2 Linear^2 Models.* Advances in Neural Information Processing Systems, volume *15*, Cambridge, MA, 2003. MIT Press.

[15] Kullback, S.; Leibler, R.A. *On Information and Sufficiency.* The Annals of Mathematical Statistics 22 (1): 79–86. 1951.

[16] John W. Eaton. *GNU Octave Manual*. Network Theory Limited. 3rd edition, 2007.

[17] John W. Eaton. *About Octave*. http://www.gnu.org/software/octave/about.hml. 2009.04.20.

[18] Octave.org. *Frequently asked questions about Octave (with answers)*. http://www.gnu.org/software/octave/FAQ.html#dir. 2009.04.20.